bawk: a versatile text-processing language

Ashley An, Christine Hsu, Melanie Sawyer, and Victoria Yang

COMS 4115: Programming Languages and Translators
Fall 2018

Contents

1 Introduction

bawk is a versatile text processing language loosely inspired by awk, building upon it with imperative constructs such as infinitely nested mutable arrays and control flow structures. bawk streamlines the process of file reading and repeatedly executing operations over lines of a file. It syntactically resembles C and other similar imperative programming languages, but features awk-like features including `BEGIN`, `LOOP`, and `END` blocks, powerful string and regex matching, and intuitive file manipulation.

1.1 Background

Although text processing languages are relatively domain-specific, they hold a wide range of use cases. awk is a widely used, data-driven text processing language that primarily performs operations on text files. Its main function is to search through files for lines that contain specified patterns and perform operations once those lines are found.

While awk is designed to be terse and is adept at one-line programs, bawk adopts a more verbose, C-like syntax. bawk aims to make it easy to read, analyze, and write to text files in a simple and intuitive syntax mimicking that of awk.

2 Language Tutorial

This section will provide details about compiling and running a simple `bawk` program, and provides some examples of useful language constructs. Full documentation for our language is included in the following section.

2.1 Writing a simple bawk program

There are four parts to a bawk program, all separated into `blocks`:

- `BEGIN`:
  The begin block is executed before the file is read, and is the only block in which functions and global variables can be defined.
- `LOOP`:
  The loop block automatically loops through all the lines in the input file, and executes once for each line of the file. If the input file is empty, the LOOP block will not execute. The loop block can use functions and global variables defined in begin, but can only define local variables.

- `END`:
  The end block is something of a post-processor – it is for instructions that should be executed after the input file has been read. It too can utilize functions and global vars defined in begin; it can also define its own local variables.

- `CONFIG` (optional):
  The config block allows the programmer to optionally customize how the file will be read. It allows for the modification of 2 configuration variables: RS and FS. RS (record separator) defines how lines in a file will be separated. The default is a newline ("") symbol. Some files may have different line separators (i.e. Windows machines will often generate new lines with a carriage return "˚" and a traditional new line symbol "".) The FS (field separator) variable defines how fields in a line are separated. Since there are a multitude of ways that files can be split into fields (TSV files, CSV files, etc.) its useful to be able to change this value to allow for adaptable file reading. The default FS value is a space ( ).

To write a simple hello world program, we'll first need to write out the blocks. Blocks are defined with their name and open/closed curly brackets. The body of the block is placed inside the brackets.

```
BEGIN {}
LOOP {}
END {}
```

This in itself is a valid bawk program– it will simply output nothing. However, let's add a print statement somewhere to ensure that everything works as expected.

As mentioned above, the *only* instructions that can be placed in the `BEGIN` block are globals and function definitions, so our print statement will have to be placed in either `LOOP` or `END`.

```
BEGIN {}
LOOP {
  print("This is the loop block");
}
END {
  print("This is the end block");
}
```

When this file is executed, it should print out `"This is the loop block"` for every line of the input file, and `"This is the end block"` once at the end.

2.2 Compiling and executing a `.bawk` file

Now that we've written a simple bawk program, let's run it through the bawk compiler. Navigate to the `bawk` folder and run `make all`. This will build the source compiler. Then, to execute a simple bawk program, run

```
$ ./bawk.sh <program.bawk> <input.txt>
```

where `program.bawk` is the program file and `input.txt` is the input file (it doesn't have to have a `.txt` extension) If the build and compilation was successful, this should output something like

```
This is the loop block
This is the loop block
This is the loop block
This is the end block
```

2.3 Simple File Reading

The simplest and most important construct in bawk is to loop through a file. A simple program to loop through a file and print each line is the following:

```
BEGIN {}
LOOP {
  print($0);
}
END {}
```

Execute the program similarly as above.

```
$ ./bawk.sh <program.bawk> <input.txt>
```

This program will output each line of the file.

Note that `$0` indicates that the entire line is being printed. Using `$n` will print the nth field of a line, so `$2` will print the second word in a line.

The language manual will cover in depth all syntactic constructs not discussed here.

3 Language Reference Manual

3.1 Lexical Conventions

Identifiers are primarily used for variable declaration or function declaration, and can be any combination of numbers , letters, and the _ symbol as long as the resulting string is not an existing keyword.

Valid:	Invalid:
myVariable	1variable
my_variable	function
myVariable1	my.variable
_	

Figure 1: Examples of valid and invalid identifiers

3.2 Comments

A single-line comment is preceded by . There are no multi-line comment symbols.

```
# hi i am a comment
```

3.3 Types

bawk is a strongly typed language. It has 5 types: booleans, integers, strings, regex, and arrays.

Booleans

A boolean is either true or false. Its type is denoted by the keyword bool:

```
bool b;
b = true;
```

Integers

An integer is defined as a sequence of one or more digits representing an integer, with the leading digit being non-zero (i.e. [1-9][0-9]*). Integer types are denoted with the keyword int:

```
int x;
x = 123;
```

Strings

String literals are defined as a sequence of ASCII characters enclosed by a pair of double quotation marks. String types are denoted with the keyword string:

```
string s;
s = "abc!!";
```

Regex

Regex (regular expression) literals are denoted with a rgx type and enclosed in a pair of single quotation marks. Regex types are denoted by the keyword rgx:

```
rgx example;
example = 'p.n';
```

Arrays
Arrays must be initialized by their type and the number of dimensions. Each array must have elements of all the same type, even when nested. Arrays are dynamically-sized. There are int, bool, string, and rgx arrays.

```
int[][] arr;       # 2-dimensional array with type int
arr = [[1, 2, 3],[1, 2, 3]];
```

3.4 Special Variables

`RS`
RS is a built-in string variable for the "record separator". The default value is a newline character, but this can be changed in the CONFIG block, e.g., RS = "\r\n"

`FS`
FS is a built-in string variable for the field separator. The default value is a space, but this can be changed in the CONFIG block. Indicating that fields within a record should be separated by commas is FS= ",";

`NF`
NF is a built-in special variable that retrieves the number of fields in the current record that is being read in LOOP. This variable is useful when iterating over the fields in a line.

3.5 Keywords

function
`function` denotes how the user starts a function declaration; an explicit return type is required. Nested functions are not allowed.

```
function int add(int x, int y) {
    return x + y;
}
```

return
`return` returns the data type at the end of a function. If there is code after a return statement the parser will throw an error that halts execution.

```
function int add(int x, int y) {
    return x + y;
}
```

true
`true` is a boolean keyword that means something is true.

```
bool flag = true;
if (flag){
  print("hello");
}
```

false
`false` is a boolean keyword that means something is false.

```
bool flag = true;
int x = 6;
if (x>5){
  flag = false;
}
if (!flag){
  print ("x is greater than 5");
```

3.6 Operators

$ (Access)
$n retrieves the nth string field in the current record.

```
$0 # refers to the whole record
$1 # first field in current record
$2 # second field in current record
```

If n is greater than the number of fields in the line (`NF`), the access operator will return an empty string to allow continuous execution of the program.

!
! is the logical not operator.

& (Concatenation)
& is used for string concatenation, e.g.

```
string s;
s = "hello" & "world."
```

&&
&& represents logical AND, e.g.

```
int i;
i = 5;
if (i == 5 && i < 6){
   print (int_to_string(i));
}
```

This will output 5.

||

|| represents logical OR, e.g.

```
int i;
i = 5;
if (i < 6 || i > 10){
    print (int_to_string(i));
}
```

This will output 5.

-

- is both a binary operator used for the subtraction of integers and a unary operator used to represent negative numbers.

+

+ is used for the addition of integers.

*

* is used for the multiplication of integers.

/

/ is used for the division of integers.

++

++ is used for the incrementation of integers by 1, e.g. x++.

- -

- - is used for the decrementation of integers by 1, e.g. x–.

+=

+= is the incrementation and assignment of integers by a specified integer, e.g. x += 5

-=

-= is the decrementation and assignment of integers by a specified integer, e.g. x -= 5

[]
[] is used to read from or write to arrays at a certain index, e.g.:

```
if (array[0] == 5){
    return array[0];
}
```

<=
<= is used for string or integer comparison and returns a boolean that indicates if x less than or equal to y.

<
< is used for string or integer comparison and returns a boolean that indicates if x less than y.

>
> is used for string or integer comparison and returns a boolean that indicates if x greater than y.

>=
>= is used for string or integer comparison and returns a boolean that indicates if x greater than or equal to y.

==
== is used for string, integer, or boolean comparison and returns a boolean that indicates if x is equal to y.

!=
!= is used for string, integer, or boolean comparison and returns a boolean that indicates if x not equal to y.

~: Regex matching
~ , when used as x ~ y, returns a boolean that indicates if string x matches the rgx denoted by y.

!~

!~ , when used as x!~ y, returns a boolean that indicates if string x does not match the rgx denoted by y.

%
%, when used as x%y, returns a boolean that indicates if rgx x equals the rgx denoted by y.

!%
!%, when used as x!%y, returns a boolean that indicates if rgx x equals the rgx denoted by y.

3.7 Structure of Program

bawk programs are divided into four blocks: `BEGIN`, `LOOP`, `END`, and `CONFIG`:

```
BEGIN {...}
LOOP {...}
END {...}
CONFIG {...}
```

`BEGIN`, `LOOP`, and `END` blocks must be present in a `.bawk` program in that order, although they can be empty. The `BEGIN` block allows global variable declarations and function declarations, both of which are visible to the scope of the entire program. Function declarations can be done in the `BEGIN` block only. Functions can be called from within the declaration of a function if they have been defined already. The statements in a `LOOP` block are executed for each line of an inputted file. The `LOOP` and `END` blocks can contain variable declarations and statements, but no function declarations. They can use any variables declared in `BEGIN`. Variables declared in `LOOP` and `END` blocks exist within the scope of their respective blocks. Local variables declared in the `LOOP` and `END` blocks may have the same names as global variables, but they will override the globals within the scope of the block.

Variable declarations must all occur together at the beginning of a scope.

The CONFIG block is optional, but must be the last block of the program if it is there. The CONFIG blocks purpose is to set the record separator RS and the field separator FS, and can only take expressions that set RS and FS.

3.8 Functions

User-defined functions are allowed. Function declarations must be written inside the BEGIN block. They must have an explicit return type (which can be void). A function declaration looks like this:

```
function int add(int x, int y) {
    return x + y;
}
```

Function declarations cannot be nested. Functions can be called from within a function declaration as long as the called function is already defined.

3.9 Control Flow

bawk has if statements, for loops, and while loops.

if

`if...else` denotes a conditional statement, e.g.:

```
int x = 0;
if (x < 10){
    x++;
}
else{
    print(int_to_string(x));
}
```

for

A for loop iterates through a block of code a number of times specified by a incrementation variable. The parentheses must contain 3 statements delimited by semicolons. The first segment contains the initialization of a counter variable, the second is the condition under which the for loop will continue looping, and the third is what happens to the counter variable after each iteration.

```
int i;
for (i = 0; i < 10; i++) {
    print(int_to_string(x));
}
```

while
A while loop iterates through a block of code until its corresponding conditional expression is not met.

```
int x;
x = 0;
while (x < 10) {
    print(int_to_string(x));
    x++;
}
```

3.10 Standard Library Features

`int string_to_int(string a)`
This function converts a string into an int provided that the string follows the regex [0-9]*. The `string_to_int()` function is especially useful for parsing integers from the input file and performing arithmetic operations on them.

`string int_to_string(int a)`
This function converts an int into a string. The `int_to_string()` function is especially useful when printing out integers, as the `print` function only takes strings as parameters.

`string rgx_to_string(rgx a)`
This function converts a rgx into a string. As above, this function is useful when printing out the values of regex expressions.

`string bool_to_string(bool a)`
This function converts a boolean into a string. Also useful for printing out the value of a boolean.

`int length (arr[] a)`
This function returns the length of the array a.

`bool contains (arr[] a, val b)`
This function is used to check if `b` is present in the array, given that `b` is the same type as the values contained in `a`. This can also be used to determine whether an array index has been assigned or not. Returns `true` if item in array, `false` otherwise, e.g.:

```
if (!contains(array, $a)) {
    array[length(array) - 1] = $0;
}
```

`int index_of(arr[] a, val b)`
This function returns the index of the first instance of b within the array a by value, given that b is the same type as the values in a. If b is not in a, return -1.

```
arr[] a;
a = {1, 2, 3, 2};
contains(a, 2); # returns 0
contains(a, 0); # returns -1
```

`void insert(arr[] a, int b, val c)`
This function inserts c at a[b], given that b is between 0 and `length(a)`, else it throws an error.

```
arr[] a;
a = {1, 2, 3, 2};
insert(a, 0, 0); # a = {0, 2, 3, 2};
insert(a, 7, 0); # throws error
```

val delete(arr[] a, int b)
This function deletes a[b], given that b is between 0 and `length(a) - 1`, else it throws an error. It returns the value previously at a[b] if it is a valid deletion.

```
arr[] a;
a = {1, 2, 3, 2};
delete(a, 0); # a = {2, 3, 2};
delete(a, 7); # throws error
```

void print(string a)
This function prints a string input. In order to print other data types, they first need to be converted to strings using built-in functions. A new line is automatically printed at the end of the string.

```
print("hello");
print("world");
# Outputs:
# hello
# world

print(int_to_string(2));
# Outputs:
# 2
```

void nprint(string a)
This function prints a string input without a new line at the end. To print other data types, they must be converted to strings using type conversion built-in functions.

```
nprint("hello ");
nprint("world");
# Output:
# hello world
```

4 Project Plan

4.1 Team Roles & Responsibilities

Each team member made significant contributions to the parts of the language listed below, but all members were in constant communication about every aspect of the language and each member worked on most parts of the compiler.

Team Member	Responsibilities
Ashley An	Compiler Front End, C Libraries, Code Generation, Testing
Christine Hsu	Compiler Front End, Code Generation
Melanie Sawyer	Compiler Front End, C Libraries, Code Generation
Victoria Yang	Semantic Checking, Code Generation

4.2 Project Log

```
commit f4bd2c0b3c3ba3bbca03696f871516f6d59eac1a
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 18 21:10:47 2018 -0500

    Update README.md

commit 7ca7038084aea88554a425a2c668f19aa0aa71b9
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 18 15:03:42 2018 -0500

    edited demo and input.txt

commit 58eb8b92e1db8c3c92df16b7efcf11c958dadc78
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 18 14:46:54 2018 -0500

    added matrix multiplication

commit 50ec7015d015b2abc819151d884124c5fdb41e84
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 17 19:20:37 2018 -0500

    created demo folder

commit d1f040521497701c46a0f3ad6295296ac4185d57
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 18 13:46:11 2018 -0500

    add shuffled file

commit a5e45a196928a3b1c96fc158ad13da3a4f2fd4b9
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 18 13:43:17 2018 -0500

    add finalized demo

commit 35899684fbffe3bb785841162315aff2a6fa2b12
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Dec 18 01:11:11 2018 -0500

    add array init to demo

commit 04248baa9f3290951e04860bc127defc01fd426c
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Dec 18 00:56:59 2018 -0500

    working on demo

commit 463408b5e171ded0de0bc1a6bd25d1f44bd5bf4a
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 19:09:55 2018 -0500

    add shuffle and demo.bawk

```

```
commit d4bf4f0c937d7d9f1888eb1a10aa9071714b2d05
Merge: 24f638a e39ba97
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 19:00:35 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 24f638a7ce46362ddf0fd4a0c7e2045296598731
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 19:00:06 2018 -0500

    add file

commit e39ba97cc337812a4b756e1620e1b9be16c2b796
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 17 18:40:54 2018 -0500

    fixed dollar and RS/FS tests

commit 95acdb938ed8e338171c6f77b585455d63498924
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 18:37:06 2018 -0500

    fix $0

commit 64aa638b0ef8d7131f7b7abad99e7ba8f259b874
Merge: 6d901cd 5abf55e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 18:29:31 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 6d901cdd1d96c2db88ccf4f3ac8acc0e4d4d6956
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 18:29:06 2018 -0500

    fix dollar

commit 5abf55e68643f46baccdda0eb7204e3afd2528b8
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 18:26:25 2018 -0500

    fix IOOBE

commit c64929f0b38f49070138d76c3217cf3f51e33392
Merge: 604987c 476fcc3
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 17:48:22 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 604987cace35ca14951cd255e5b25e4c6d4cf271
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 17:47:46 2018 -0500

    fix $
```

```

commit 476fcc35116efd48c6256ced67781f45d9ca86c5
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 17 17:32:59 2018 -0500

    fixed formatting of test

commit 921d0913bb3ec5a638ba73c10daad5bc0fa88b1e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 17 17:29:45 2018 -0500

    switched RS to FS

commit 2b114337cfe8e6015469ad9fbfee8ac2933f4b7e
Merge: c6870be e1c358f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 17 17:28:13 2018 -0500

    merge

commit c6870bed867f7a89f4753022fb5d0d24798698ad
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 17 17:25:24 2018 -0500

    cleaned up and commented code

commit e1c358ff49b5524a542b4575319f4644e077ad76
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 17:23:22 2018 -0500

    add file with columns and constitution

commit a77c6858cc8fee5c9b4e700bf05baaa89aa8e513
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 17:17:33 2018 -0500

    add shuffle.bawk

commit a737c26dd43768af98e908c0c4a7f5c6b6efcc94
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 17:17:05 2018 -0500

    demo

commit a741435266c3f32ebf6aa00edb35826efd193094
Merge: 61a8e87 94e6b02
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 17:16:42 2018 -0500

    Merge branch 'master' of https://github.com/soybean/bawk

commit 61a8e87fe4b9fb462ebd43738a932f211039e78f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 17:16:34 2018 -0500

    add nprint to codegen
```

```

commit 94e6b02c8cc85cc30eb7a356613f868d97005d90
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 17:13:45 2018 -0500

    add constitution to ascii art

commit ec2c9cf5aed89d69de026df897a6fdad29ce0529
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 17:13:18 2018 -0500

    remove comment

commit fd46c48df455b0e6195929f86750ea2afa7eb004
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 16:59:17 2018 -0500

    add the constitution

commit caba497b9b6e2c324d4e3bba76b1922f8f202691
Merge: 7b18691 8e03324
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 16:59:06 2018 -0500

    Merge branch 'master' of https://github.com/soybean/bawk

commit 7b18691a22fd138c305acc787ce30907fbfe56ad
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 17 16:56:37 2018 -0500

    add the Constitution

commit 8e0332407b896fb0c97d72d59b945f11ebb135b3
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 17 04:37:13 2018 -0500

    add demo text file

commit 95a328eb1727872a2b15ca6e7e29bf7f5ac440fa
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 20:17:21 2018 -0500

    removed unnecessary script

commit 842fdc02b05249011ec983d6adb559c58452e179
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 20:16:34 2018 -0500

    cleaned up files

commit b1d2645cf1975c964da82830a51db5bb2a487b48
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 20:13:30 2018 -0500

    added .out and .err files for tests

```

```
commit 55cf1296ecd4f9198db83e69ceb735e2256601f8
Merge: f02e5de c183cd1
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 16 19:59:37 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit f02e5de67b16b2f0062a58a4e99996d388a658f0
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 16 19:59:18 2018 -0500

    remove warnings

commit c183cd1d668c600c2b211cae8a5a5c4cc3fa9937
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 19:54:22 2018 -0500

    fixed testall.sh so that it prints both stderr and stdout to .out file

commit 6f4c79e1874f7bf983ba86ea91955e19429a3b64
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 19:39:15 2018 -0500

    removed test and fixed generate-tests script to print out both stderr and
        stdout to .out file

commit 2cff306804050f5e8a81fcbfd07b396d1972a7c7
Merge: 97c1825 4ff24ee
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 17:54:06 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 97c1825ea42294bd096962a20eabebf77da1e1cc
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 17:53:56 2018 -0500

    moved runtime fail tests to pass tests

commit 4ff24eef455ff8baab198e2c4b82bae2c9978692
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 16 17:41:21 2018 -0500

    remove print statement

commit 15cc960c0bf7d6b129693ae39fb859b56d599964
Merge: e7156d3 ca1d52f
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 16 17:36:42 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e7156d39730099674cf5fea3b32ab6e2834776ec
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 16 17:36:25 2018 -0500

```

```
    add RS/FS

commit 3952cf8cb41f71d4d88b7db408397f3461dd3a3d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 16 15:39:31 2018 -0500

    broken stuff

commit ca1d52f91265627f72766b037592e18d321517f5
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 16 14:00:13 2018 -0500

    print usage message to stderr

commit c4ecf6208c39083d7da2b6c608be4b492d55e230
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Dec 16 11:04:16 2018 -0500

    print error to stderr

commit c3eff0255b80f756ead323351ec74e4563aee4c4
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Dec 16 10:29:40 2018 -0500

    index out of bounds error

commit 04f5f3de3e06be976dc97dd9d0e83825258a3b78
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Dec 15 19:58:37 2018 -0500

    add index out of bounds error for arrays

commit 404b40317d70797f90174aa49faa3da89623c16c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 23:55:38 2018 -0500

    added usage message for when input file isn't given

commit 82d8aa5b9a401e5022faf49e4c38c1e8a3cd0597
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Dec 14 16:30:30 2018 -0500

    fix bool operators

commit 1adf3ca74b42950e0ae0f2baec15d253d9ef2814
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 23:39:06 2018 -0500

    fix syntax

commit 5f54e2a52e0611ef924aac9219277707d86ae518
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 23:37:18 2018 -0500

    added config tests back

```

```
commit 00593446fcdca8364f5b09eb019b020138da8532
Merge: 95026ff 32ed67f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 23:23:41 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 95026ffd23e441f3b9aacf110f1d5501d8143336
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 23:23:36 2018 -0500

    creating testing suite script

commit 32ed67f5d4b320b706c71a74324f11bfe0f0e065
Merge: 12bc74e 47c6140
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 21:14:19 2018 -0500

    fix merge conflict

commit 12bc74ec1e4577cbad2a2fdfed874d4cda0e833c
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 21:13:30 2018 -0500

    attempt index out of bounds check

commit 4bdaff44e9f85c7bda2aa1d954a0fd0b78f4da63
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 21:13:04 2018 -0500

    attempt index out of bounds check

commit 47c6140b1afae2be6aaecf2e9f1f997470173b9a
Merge: ebce545 94ac77d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 21:10:17 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit ebce545674f428b06f0632cb9c42d347ddde764d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 21:10:06 2018 -0500

    fixed formatting

commit 94ac77d41f7853861fd7ec5c64cd02128d10009b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 20:52:39 2018 -0500

    renamed test.txt for demo

commit f8f2bc92e4d2a7a1ce13468c93eeec5ce913f002
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 20:33:51 2018 -0500

    removed test
```

```

commit c5afa988328574c041068f6db8145ee408a99b79
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 20:32:59 2018 -0500

    fixed formatting of test

commit 2687261e2d6fcda9d446839898e6d8665cb55c81
Merge: 5ac0cf7 19010a6
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 20:27:53 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 5ac0cf76d5453cf84fad6da6dba7017d835b6dd7
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 20:27:39 2018 -0500

    fix delete for arrays

commit 19010a6ead0ac3688c0a8ab53f419345cdcea7c9
Merge: 5d298c6 9a575a8
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 20:26:18 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 5d298c60a3290ab22c39f7c563a31e35359be53d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 20:26:14 2018 -0500

    fixed dynamic arr tests

commit 9a575a880e0f73d7bb51423ff69df3b03ddb1855
Merge: 0804277 e2fb5a2
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 20:19:38 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 08042770bed615dc896164550b1770478cd08737
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 20:19:18 2018 -0500

    remove weird files and add stephen edwards in ascii art

commit e2fb5a23518904b121912df866bd60c9b0c06d0d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 20:16:42 2018 -0500

    get rid of unused code in semant

commit 6ee28f6f6492ccfe7a9919a6946ccefae8075048
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 20:14:26 2018 -0500

```

```
    deleted config tests

commit a52c1f116d43236dcbea3eaa2d6ef7270c486827
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 20:04:33 2018 -0500

    fixed delete in semant

commit 33717d6a96c9ee10a7de91175f245ca97d0d77c9
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 20:03:23 2018 -0500

    fix remaining warnings

commit f4b13394088b1958b23b32d4f90f80b156045e17
Merge: c12c0ad 15fe3ee
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 20:00:45 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit c12c0ad894183167e6c42172528eeb6afa56505b
Merge: 32b0e72 73fd684
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 20:00:22 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 32b0e72a20cf83444c9ed377b3904fb8a5546ab3
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 20:00:04 2018 -0500

    get rid of some warnings

commit 15fe3ee3825f5355970104bf9dc0477e17e1d687
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 19:58:22 2018 -0500

    Clean error messages in codegen

commit 73fd6844301f12dc8df7ec86b305b95b16ecadda
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 19:56:26 2018 -0500

    fixed for2 test

commit 3da5d6ee016f42470810cc59652349d832f2492a
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 19:45:59 2018 -0500

    fixed loop/end name

commit 862608b1a3ab13946a6aea54a45e33d70bf7280f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 19:43:14 2018 -0500

```

```
    remove comparelists

commit cb61dbe1e1d0cf0ea1a5d466077e65413c01b7f2
Merge: cdddf5c 4ee85f1
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 19:37:22 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit cdddf5c3f26cb727dec8f80dbe86a011be8b6248
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 19:37:11 2018 -0500

    add index_of

commit 4ee85f18c9069d220d39dbe53a141f2b93b73d7e
Merge: 3bff4ad 12393e0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 19:33:47 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 3bff4ad39c7730d40879b9fca150ab2415ad76c8
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 19:33:43 2018 -0500

    added loop and end fail tests

commit 12393e00f29c2ba7ce23e7baa952da58cec1d39b
Merge: a71afd6 031bc22
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 19:30:14 2018 -0500

    contains works

commit a71afd646ad10aa2fef2755797eaeeeec9ab7540
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 19:28:47 2018 -0500

    contains works

commit 031bc2200b984a880b3e90074c7f92ccb4b689db
Merge: 71b0bea bdef67d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 19:21:37 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 71b0beab990e85d27aba51a36bc054aa5d61e91d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 19:18:29 2018 -0500

    fix some warnings

commit bdef67d70875919c0ec76d3a01b3ec1db73a900a
Author: Ashley An <ashley.an@columbia.edu>
```

```
Date:   Thu Dec 13 19:17:04 2018 -0500

    removed main function for mylist.c

commit 499b4d0c57377fbaea5f88e95b406cbb6e36e3fb
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 19:08:42 2018 -0500

    fixed comparators to take in unsigned longs instead of const void *

commit f98ea40080c7b81d89fd0236c439f48f1966f994
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 18:44:37 2018 -0500

    edited README

commit 993c72da72b557b84f05ddbf6d8e266360ed3c79
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 18:42:45 2018 -0500

    edited README

commit c5fee69b7b3202a3ff534be78c3cd72e34f780d9
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 18:41:30 2018 -0500

    edited README

commit b98fb56c0d353118019b2777ff3b98d76f5d3419
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 18:40:06 2018 -0500

    edited README

commit 3e2118083ba7eb904dac22df11eab72fc9e9bc26
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 18:37:50 2018 -0500

    edited Makefile and bawk.sh to be cleaner

commit 650b325678c96123caef7b6f1acd828a9403c230
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 18:27:04 2018 -0500

    removed unnecessary script

commit 145d9b3aad73a42c60a9c4d77e0f022ccd7b1817
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 17:35:07 2018 -0500

    fixed contains and indexOf

commit 89951de56381258d31dd80878997666f17587003
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 17:18:03 2018 -0500

```

```
    fixed bawk.sh script so that it stops execution when there's an error

commit df46c68f6c444e23ada5ab81375ef2a2ea4fda71
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 17:09:18 2018 -0500

    fixed error message for insert in semant

commit af4295347fa2dce81ea25156f44d331119d6ee3d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 16:56:47 2018 -0500

    fixed minor error in semant

commit aba9e1e6aafba41be6f779dea620ceedae8ec303
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 16:49:23 2018 -0500

    fixed formatting of test

commit 47a37c89b9c4b3020c0a08f5f6503c0fc7c23bb2
Merge: 816ea6a f41973a
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 16:38:50 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 816ea6ad2f05602e44e70af763ed5cb2eb78fc55
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 16:38:44 2018 -0500

    fixed regex assign test

commit f41973ae64d3999647fb04752ece039ddef36e82
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 16:36:28 2018 -0500

    Update README.md

commit f3ec5b4a8a0f2042f0756e6f52203a8479beaa11
Merge: 862afa9 676628e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 16:29:18 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 862afa9a9c970be2d4b83ec41b44078f4df82c50
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 16:27:55 2018 -0500

    add single run executable

commit 676628e9e0faa344b65a125f7de15fe077d68a4d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 16:08:38 2018 -0500

```

```
    cleared warning

commit ca7e71d700af84d7ed3876eb331e4a864d724bcd
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 15:12:17 2018 -0500

    clean up warnings

commit 80fd93779db4f8f8de2e9f43159ff8aeaa64e22d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 13:55:05 2018 -0500

    remove print statements in test script

commit e3d020bba6af80ece31ca077d5e25dbd854cfad6
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Dec 13 13:30:28 2018 -0500

    string operators

commit 6af4508e5586421e3cfbe409a2c4b020b5bd98a2
Merge: 061167a 9884e63
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 12:33:55 2018 -0500

    merge

commit 061167ac79c78f9c98cf844dadb77322011400da
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 12:28:09 2018 -0500

    clean up warnings

commit 9884e6372114d368a0350ce648decdedd4c98190
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 12:19:02 2018 -0500

    cleaned up Makefile

commit 7c1789f565b65c3a2dca24c66855f23b8abc15f3
Merge: 7d57972 75f957d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 12:12:59 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 7d579729c6faf36a9ab73edd39bc3438b382207e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 12:12:53 2018 -0500

    fixed make clean

commit 75f957db467c701718dd2b8d1739d4b9da8ca54b
Merge: 9b74c90 2d61756
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 12:11:40 2018 -0500
```

```

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 9b74c90bb0a998e51d51a1b8b81d61fd86781673
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 12:11:27 2018 -0500

    take care of some warnings

commit 2d61756b46c4c6347df15b7f63b3bddd28adf8ec
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 12:04:22 2018 -0500

    removed unnecessary script

commit 3afa53dafa6690f8ab9dde07b83ef1b5c51d6298
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 11:43:52 2018 -0500

    removed unnecessary test files

commit cf666ef3f8b8f737f342c9d7a16c0a5f8efa5f8d
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 11:37:56 2018 -0500

    add rgx comparator

commit 15f319f491df1d4f680d8b6b7ee491ff12217f68
Merge: d4b02bf 1e5c8c6
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 11:30:04 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit d4b02bf90a0d50e0222f81c3d5d7ad2a6b687b5b
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 11:29:55 2018 -0500

    index_of works for string arrays

commit 1e5c8c6af3e064027a2af637169a8f9f66598409
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 11:29:38 2018 -0500

    fixed formatting of tests

commit 508cc1f5375d1a83edcc8ce2479be95a353aeacf
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 13 11:28:07 2018 -0500

    edited contains tests

commit 2f82fb9bf24352f8876e115f2d71a58d1199368d
Merge: cabe654 805e444
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 11:22:00 2018 -0500
```

```

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit cabe654c1caf93c570d60dea93a933cbe9f1965f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 13 11:21:41 2018 -0500

    arrayderef assigngit add codegen.ml

commit 805e4447089781ee430e9bbba89ee8911c5ed53c
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 10:59:12 2018 -0500

    contains works for string arrays of any dimension

commit 734f0fe20524437c3eaec7a8fbb6c700b5e2e4f7
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 10:51:52 2018 -0500

    Use legit comparator

commit d8079d52090d01419c92a951121a6ff203932130
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Dec 13 10:46:01 2018 -0500

    string contains works

commit c0cadedc3299c2911602d024ef9972f0b3b05a8f
Merge: 10b7d42 02675a1
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 23:19:59 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 10b7d423c8c93706ab0bef99b9d04305881d7002
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 23:19:46 2018 -0500

    change ret type of contains

commit 02675a1f5ab975ce12d5804ef9164dd3c75e98ba
Merge: 1b4b93c 4731ba7
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Dec 12 22:10:19 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 4731ba7ef29ffe719702cd19806fef6675a0c886
Merge: 6fce76e 86952f3
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Dec 12 23:01:00 2018 -0500

    add string stuff

commit 1b4b93cdc6a72221fb888252a89278fc8caeb7b3
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Wed Dec 12 22:10:07 2018 -0500

    added rgx

commit d04dddc7c73adf93a41bbb25f3f23f34277336e4
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Dec 12 22:08:35 2018 -0500

    Revert "added regex and working on assign"

    This reverts commit 5bff3248402a86d4a62f3a89c98dce8536c9a409.

commit 86952f3d8b8bbc470558361ddc2f0858743deb4b
Merge: 5bff324 5a4c312
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Dec 12 21:56:16 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 5a4c312be451c193253a42f6f6ba4d164125237a
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 22:37:02 2018 -0500

    fixed formatting of test

commit 57a93defe69ea159ce89c089b96751ae5ceddcf3
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 22:35:32 2018 -0500

    added increment back into refactored codegen

commit 5bff3248402a86d4a62f3a89c98dce8536c9a409
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Dec 12 21:55:59 2018 -0500

    added regex and working on assign

commit 8521a7683aa142b55c8bc052b787838451a181cb
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:31:12 2018 -0500

    fixed tests

commit cc629562d8614f8941ea37be3f9619eae288b1d6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:28:27 2018 -0500

    fixed tests

commit ec03b397940122ee247337e3737057741289fbf2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:15:05 2018 -0500

    created a variable called rs to hold RS that we want to string from each read-
       in line

```

```
commit a3d61560e223bcb8913e43f17e181581dd9b0913
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:13:07 2018 -0500

    fixed dollar so that it now strips \n from string

commit e5541e7306243746d3387579c9d2c733f55e8ea2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:04:44 2018 -0500

    deleted redundant test

commit 6b770e69952d9f027d429ffc538293a37536c0ae
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:03:10 2018 -0500

    changed name of test to be grouped into all dollar tests

commit d67e45e118f790c5741197a2bd1a5c8fe0d2947d
Merge: 7430a02 3544e38
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:01:40 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 7430a02516c30e10343e67d0065fc81f484b2aa2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 21:01:26 2018 -0500

    fixed config tests

commit 3544e386e65f9b4c2310cab1cb5b8993e63b427e
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Dec 12 20:49:45 2018 -0500

    set assign

commit d5ec0b319bd3d44322ac7d181c30a91ae3a8d2e3
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 20:49:23 2018 -0500

    changed True and False to true and false for bool_to_string

commit fc47e0d01928d0fb8feefde14a1c434ee2e9a4f9
Merge: d26dd13 e62a16c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 20:43:44 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit d26dd1351469f57d856b9baff5f9296c185cf1e9
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 20:43:40 2018 -0500

    fixed tests

```

```
commit e62a16ce2ec59ca74a1c1de7f36c8dfa405b1faa
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 18:21:56 2018 -0500

    array_test

commit 41b0fa6816a06dd933432c516421f66224cf9fe0
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 16:06:16 2018 -0500

    contains changes

commit 6fce76ec318c617b61d1d6a57c8140208d550185
Merge: d221056 a97e23c
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Dec 12 15:47:52 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit d221056d363394b3bea2a31aa3e261dfbc6b033b
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Dec 12 15:47:21 2018 -0500

    equals

commit a97e23cd01fb5616a55b2c2552045536d2ae56f5
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 14:46:11 2018 -0500

    new contains attempt

commit ad6d4d1f8c6066591d536d73bbbabc5dad0f3ed1
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Dec 12 14:14:08 2018 -0500

    created contains and indexOf wrapper methods

commit 2597b15b8c0721d436487c814986db881d89841f
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Dec 12 13:26:45 2018 -0500

    add rgx_to_str

commit fa32cbacd614948091037b4c1bfdaefa44e71123
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Dec 12 13:18:11 2018 -0500

    fix NF for empty lines

commit 9e1c2f1ed7da2f85a37c11886748bff9a3cdb61f
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Dec 12 13:16:13 2018 -0500

    add NF

commit 38f857cd74ab392a9784ba60bb40131840c3e807
```

```
Merge: c071142 49cb85c
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 12:17:30 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit c071142dbb89e8e62ab89964bfe6efd5ec7a2a0f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 12:16:55 2018 -0500

    add insert, delete, length built-ins

commit 49cb85c713e9129332feb290b48f5c2f9b2a59dd
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Dec 12 09:02:37 2018 -0500

    clean indents

commit 4f8e25d5cd5cdf8d96872d55105261f7b33e452e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 23:32:50 2018 -0500

    removed test

commit bf23c9fc69175bf06661c5eb01d3dc471796feb9
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Dec 11 22:57:48 2018 -0500

    Add arrays to restructured codegen

commit de9f16e04b4aa2a7ec93ca226c57b1b1a0d340c2
Merge: 5871087 56812fc
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 22:29:55 2018 -0500

    Merge pull request #13 from soybean/ashley

    Ashley

commit 56812fcd486cbb631bd28c13b52ec8bb8c3db379
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 22:28:50 2018 -0500

    added other stuff from loop and end back into func

commit ce0d72f2ff5e80d206874beb55c6bfc06fba1e06
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 22:07:39 2018 -0500

    fixed assign

commit 119b59927bb0120809949220be4473529367fa60
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 21:54:22 2018 -0500

    if, for, while work in loop and end blocks now
```

```

commit a1d5f0a99855f95a7250fca03902f5655d141ca6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 21:53:40 2018 -0500

    removed preventing dollar in all functions from semant for now

commit e9d4aa9bf961722c4fe12207446a5118fc3aa4a1
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 21:27:47 2018 -0500

    if statements now work in loopgit add sadtest.bawk

commit bed0025852a7bc2bf424972a6c01782236220018
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Dec 11 21:06:39 2018 -0500

    edit function declarations

commit 845738dbe7435d30576d979581960362450184a4
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 20:04:57 2018 -0500

    refactoring codegen to make loop and end functions instead

commit d7900c2c87cbe14ad24ed9bd622503f19a049327
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 19:15:00 2018 -0500

    edited sadtest.bawk

commit a07f5040569f24e707bf610d972b1cdb569a405d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Dec 11 19:14:46 2018 -0500

    turned loop and end into functions in semant and sast

commit 58710879080b2b98ab5724c57a7f21651636680f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Dec 10 22:06:03 2018 -0500

    Delete array.s

commit 564530b48cdc6385741e160ed054205d91b70058
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 10 21:19:29 2018 -0500

    edited Makefile

commit 185fe79e119bd3a8eec97a87f2025f2a2067050f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 10 21:18:54 2018 -0500

    changes to Makefile

commit cdc6775a0b424e3a868542fb7081b078621ecfa2
```

```
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 10 21:16:24 2018 -0500

    added compareStrs method

commit 80e47b54181334a49f863a7e66223b38bf589b2f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 10 21:15:41 2018 -0500

    added compareStrs method

commit bbfb7528a9d3a3df39dea6400fc7f7a3e621e9b0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Dec 8 16:50:33 2018 -0500

    added contains method that returns int

commit 13d48e879a8532fe759a6ed4f9d9a3c27b5b848d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Dec 8 14:31:48 2018 -0500

    testing loop block if statements

commit dd907142fdf448670e95d48f77e96748329f0568
Merge: 9009905 3bfc962
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 14:54:12 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 3bfc96283a5ff36e0dab992fd1f820c92f099480
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Dec 7 12:42:38 2018 -0500

    update begin with some exprs

commit 900990564128b5f077dcff88d066b1e33bbba4f0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 14:54:04 2018 -0500

    created string_to_int test

commit 9ecd96bc4d88d14e7765dea692573ca12de91eab
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 13:58:42 2018 -0500

    fixed regex test

commit ab738de82bdf9ab4837731efd32170b7998a1193
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 13:56:57 2018 -0500

    fixed regex tests

commit 8d333ca1710cb03b8d842bfd3ff8ec8d1c9394f7
Author: Ashley An <ashley.an@columbia.edu>
```

```
Date:   Thu Dec 6 13:44:50 2018 -0500

    fixed increment and decrement tests

commit e95791136a92b4f1187912760d162df400468dd9
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 13:25:46 2018 -0500

    edited makefile clean to remove .o files and removed duplicate int_to_string
        in convert.c

commit 7a7756fffb5fdc5437ea1ddbdb24a0b413e68ca8
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 12:09:06 2018 -0500

    made dollar test print to test it

commit 6832811e2d0cb840c8f874486b56654de0bdae1a
Merge: c719e4c b966e03
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 11:21:29 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit c719e4c56534ad015e18e9f99ccc2bd4540840ed
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 11:21:25 2018 -0500

    added test for rgx to string comparator

commit b966e03845e8e4c3e4a1ef4f7a00aeee1083d2a5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 6 11:20:28 2018 -0500

    add semicolon

commit 74fa4e51720a450e1e00884d9fcfdaf0b79ca299
Merge: f375848 416bbde
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 6 11:16:51 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 416bbdee3ec404990e9ba115bb11380ef30c0198
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 11:16:38 2018 -0500

    fixed regex tests

commit f37584837cd66ecb07ca4d80e6a3fc1c07e39399
Merge: 804e0e5 00ce943
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Dec 6 11:16:15 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

```

```
commit 00ce9434af2f1a295a835e5e322a0c442d6fdb73
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 11:13:33 2018 -0500

    fixed regex tests

commit 7e4a214dc125601230d9516b3b602f1e78c7d8db
Merge: 73d8f15 36c3e0b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 11:05:10 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 73d8f152459f98aa1d333a79fa22a81767d497e2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Dec 6 11:05:05 2018 -0500

    added a bunch of missing fail tests

commit 804e0e53cfff619936db3a3f831b9f57b52a30de
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Dec 4 16:12:01 2018 -0500

    need this for hw

commit 36c3e0b030a21fa51e74e2c1ae99d5b7ae39e8e4
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 15:19:50 2018 -0500

    uncomment string to int

commit e30c4be45c44b6653bc266caca1c7327c9bdff60
Merge: c95edb0 6ae6907
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 15:14:21 2018 -0500

    Merge pull request #12 from soybean/regex-stuff

    Regex stuff

commit 6ae690784fcd42384dfbee3b2dc50cd5172a4083
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 15:13:34 2018 -0500

    add regex equals and not equals

commit 26739dbc4939f7cd11558ef0c549a9048382c849
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 15:03:43 2018 -0500

    add not comp

commit 2c674551855ed1d03a80481db9532c351e8c1f6e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 14:47:16 2018 -0500

```

```
    change regex function name

commit c95edb0064ca2a4820b5831d0d52e7ac0a4418e9
Merge: 04ab7a5 c84941a
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 14:40:38 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 04ab7a586c5bb80a1896b3dfc30d6eb154aa6e94
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 14:40:32 2018 -0500

    add to gitignore

commit 8e958170a46fbc1b33b1530936c6cce6d14e1080
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 01:36:47 2018 -0500

    remove leading quote: NEEDS FIXING!

commit 4b01498dcebfed58284901e8cee7c7c17ac4f909
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Dec 4 01:26:16 2018 -0500

    add rgx

commit c84941aaa1eefb81c63a07b24e1f4ceb4ff9de8c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 3 22:58:31 2018 -0500

    tested 3D arrays

commit a077631ee32c81c5ae9a8bd35ac8104362997e47
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 3 22:50:50 2018 -0500

    removed main method from mylist.c

commit 1ddfac8fca9fd9ba3383d7b19f353a378a4fe95b
Merge: 2d1023f dffa5c5
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 3 22:48:33 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 2d1023f166678d6c3f9dd5833168ad1b93bd7afa
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Dec 3 22:48:29 2018 -0500

    changed struct Node to have unsigned long data instead of void * data

commit dffa5c501217969a8d6e7a6fd142b35dc29643dd
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Dec 3 22:19:37 2018 -0500

```

```
    codegen pluseq/minuseq

commit af72c36774581a770ab5a538250e8b4bbc1f1154
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Dec 3 19:26:31 2018 -0500

    Update scanner.mll

commit 03411832dca0eb6b1ac20d54f5dc3b3d4222881c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Dec 3 16:38:26 2018 -0500

    change insert return to be oid

commit c58b5568cc86dc71d6ac28c92ad547745958838e
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Dec 3 16:32:12 2018 -0500

    fixed test so inc/dec is postfix

commit a3b6f32c8442944ce42033ecf135a8d7358bb1dd
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 3 15:48:06 2018 -0500

    get concatenation working

commit 93f3902bc67b0946baceb8655ae1fb867f066c1a
Merge: f8376de 3d0351e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 3 15:18:48 2018 -0500

    fix merge conflicts

commit f8376dea35daef27931eb775cb45769c1143e3b7
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 3 15:14:14 2018 -0500

    add bool_to_string built-in

commit 3d0351e9a30433ae323d112e1b780cbd1679066d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Dec 3 00:47:32 2018 -0500

    update codegen to work with sast

commit 998511f3a39510908c1e3675abe6191faece98f1
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Dec 3 00:36:35 2018 -0500

    fix inc and dec

commit a83568780a96545530e4b1dc14c6ac218ec3e5b1
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Dec 2 23:42:01 2018 -0500

    revert
```

```

commit e470159b451688f8c7a7f420a308503981dec16f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Dec 2 23:36:04 2018 -0500

    made config clean

commit f83a55b16090cc74e66512208e7857f980788401
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Dec 2 23:29:09 2018 -0500

    fixed config block

commit d5d7964b3e4a3c562bf0ca4cbd1b05bc1640f779
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Dec 2 22:56:44 2018 -0500

    working on config

commit 7f818017d307337d82a3c572d741c4d1f9c30cc0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 2 20:46:54 2018 -0500

    starting testing 3D arrays

commit 4705d0c7b2ee53aa443ddc6e244dcc4bff2ac2df
Merge: 958dbdb 7df597d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 2 20:12:10 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 958dbdbac8411864adab5d9b31bcfd64605c779d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Dec 2 20:12:07 2018 -0500

    fixed compareLists method

commit 7df597d9dc682128d78f26f3939fd1af5ccee7c8
Merge: a5fb40a 9f989ae
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 2 20:07:40 2018 -0500

    fix merge conflicts

commit a5fb40a073f0e373dbe086776f114f4c5ebce673
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 2 20:06:48 2018 -0500

    get increment and decrement working

commit 9f989ae9d24a1c00083209c6d6af40f8e997b693
Merge: 6729dc5 0be4872
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Dec 2 19:31:59 2018 -0500

```

```
    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 6729dc52f9a2d3c1fd2e4e13fc27e98b93f8cd06
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Dec 2 19:31:47 2018 -0500

    fixed dollar sign in end

commit 0be487229986b8a22e27044ce223d710101c5ca8
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Dec 2 18:54:02 2018 -0500

    errors caused by merge conflict

commit bbce643c0bc66c6572d2b0f42d2662952427ea22
Merge: 42363f5 2b35569
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Dec 2 18:51:29 2018 -0500

    Merge pull request #11 from soybean/assignbranch

    assignbranch: 1d array literals working

commit 2b355693ac6d516e25a587770ddd0506172876dd
Merge: bffcba6 42363f5
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Dec 2 18:50:52 2018 -0500

    Merge branch 'master' into assignbranch

commit bffcba6312e6449a914da044ff8874ad94a01a08
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Dec 2 18:49:00 2018 -0500

    add array_test, works with 1D arrays

commit 6f175c9b99aacaa1fa4c7249e45c5dc5344e51e8
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Dec 2 18:44:06 2018 -0500

    fixed locals in codegen

commit 42363f5017bdc15bc22d7dcf46436dbbac02311b
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Dec 2 16:40:11 2018 -0500

    add increment and decrement

commit 1366da86174305701b1a8cb24100d62c3e4f16fa
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Dec 1 23:49:44 2018 -0500

    attempting to fix locals in loop/end

commit 0fb4c78d61cc60dd2c3d0eb35bcad81a974809f4
Author: Christine Hsu <christine.hhsu@gmail.com>
```

```
Date:   Sat Dec 1 17:56:46 2018 -0500

    fix scanner quotes and other small fixes

commit 4a4867b1e37c8530cb4e61b683d6d5a93f5060d5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sat Dec 1 17:47:53 2018 -0500

    rly hope empty array assign works

commit eb9b798f0d0c123290a6bb53d699c44e4cea0fec
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Dec 1 16:57:59 2018 -0500

    now can create nested arrays with linked lists

commit 136f327c224142326a06c40c60b1e310347e55f0
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 30 16:20:59 2018 -0500

    FIXED EMPTY PROBLEM?????

commit 842c179c8e67740010cbc13fd59f4b02b17bb188
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 14:11:23 2018 -0500

    pushed indexOf test in main but it's not working

commit 49d2f846768620a72e881e48d74c4780548d831b
Merge: 89ed76d 7051fe2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 14:09:29 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 89ed76dd639bbb4c4bab2f238089c3ba72ca70f6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 14:09:23 2018 -0500

    added ability to initialize and populate a nested array

commit a72d977a11920d8af5d051d2f94aa3bdd7c351ed
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 13:33:02 2018 -0500

    fixed code to not take in pointers to positions in list

commit cfab24d4fcf5c8a98723a8fd0d17eb3117a4602a
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 13:18:45 2018 -0500

    create copy of working arrays so far

commit 59f8fe9193c572b62ab41437529fc3d75fdc1d1f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 13:14:42 2018 -0500
```

```

    made it so you give actual values not pointers

commit 54196dcb3a310f508d43f9f1ba221697d03c268b
Merge: a5bccbc 5f0d948
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 12:59:49 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit a5bccbc4701f687dc4f26020148daf7ca7a58251
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 30 12:59:45 2018 -0500

    got rid of global variable

commit 7051fe21e52c9576889aa32ccdb851dbfe37501f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 30 12:55:43 2018 -0500

    changed literal test from fail to pass

commit 3c0a0f9d576d83e8fdb31683f00120e8adf3ab34
Merge: 7085f44 a72d977
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 30 12:54:05 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 7085f447ba9263fcdf725c8b3f5540fb175e07d0
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 30 12:51:36 2018 -0500

    fixed not equal in arraylit

commit 5f0d948cb79c3d75ccf06d139db11e0b54689173
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 30 11:34:32 2018 -0500

    added tests and cleaned up code

commit 131106371a19f3b4ed4188033fd68d3d450d59d4
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 29 22:27:09 2018 -0500

    fixed make all

commit 8f39826475c46890f893937443caa761fad5e3d0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 29 22:25:43 2018 -0500

    edited main so there are no memory errors

commit f2425927ce589da9a21974d8675dea404a5cf65b
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 29 16:47:47 2018 -0500
```

```

    refactor enhanced for

commit db1c2c434a792b51b11adab9b7d51afbf3485193
Merge: 7802bf1 7cc49bf
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 29 15:20:32 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 7802bf196612121c5a924ed160289c43e5b025dc
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 29 15:20:26 2018 -0500

    tested that you can create nested arrays

commit 7cc49bfa752aa356d8a5099744d29497a9e1cb8a
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 29 11:59:53 2018 -0500

    fixed syntax error

commit e4b6081411bb0c4bbaf375f1db20008867e488f9
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 29 11:57:12 2018 -0500

    refactoring call

commit efc8f6df905308e1065334bbaf5c4143167d9c90
Merge: 1c8281c 99538eb
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 29 11:55:01 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 1c8281c0ba70a4da0eccc4b78069d355c58efaed
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 29 11:54:42 2018 -0500

    refactoring

commit 2f22a64259d855e7ec45735467e4e1360266f48d
Merge: 8a21715 f6c5d01
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 29 11:41:18 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 8a21715302e2cca40924985e6a137f14578adc75
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 29 11:41:04 2018 -0500

    revert

commit 99538ebb8b285dc80a7a61c20474c962c982e595
Author: Ashley An <ashley.an@columbia.edu>
```

```
Date:   Thu Nov 29 00:52:04 2018 -0500

    minor typo fix

commit f6c5d014e3538c88e9060549100e6f419f0de5e4
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 28 21:21:21 2018 -0500

    created more files to try to implement generic methods for nested arrays

commit 8a4d23e1f3fa9b35c7d5677f3d6372905267c602
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 28 21:14:18 2018 -0500

    made init return pointer to list, made contains and indexOf generic for
        primitives

commit c166d3760e8d2e283254d63d562787cbe21e18de
Merge: 4356679 081d736
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 28 18:52:21 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 081d73614bb13ee3483c492acfb8b19247aa0c8f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 27 23:24:55 2018 -0500

    strlit quotes, escape chars in scanner

commit 435667944f1bf86cba2e70ebdb7c0d73076af6e6
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 27 18:38:14 2018 -0500

    Revert "refactoring"

    This reverts commit 6e01d637372e96b1d2e3b7a520663dadbcbfaccc.

commit 6e01d637372e96b1d2e3b7a520663dadbcbfaccc
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 27 18:33:12 2018 -0500

    refactoring

commit c38fdbab4fd8dd2ce9f210c11fa9dfda99696c37
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 27 18:27:51 2018 -0500

    updated error statement

commit df794b9bb5a1006fa45ecda89aa099b7eb14d9ee
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 27 17:25:29 2018 -0500

    adds prettyprinting

```

```
commit 2d331588f6f6318cfb092c3c9ac33ce512a669d1
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 27 17:10:44 2018 -0500

    change arrayderef in loop

commit f9f37a92d2c155d9101ba3cd1ba9f773ce3c2220
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 27 16:14:25 2018 -0500

    changed deref

commit c935b816e004eeae18195a1c85b7da089bbeef88
Merge: d0e2b6f 4cbb8e0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 27 00:15:41 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit d0e2b6f10ad12b4f4ab66d58339b71774f1f739a
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 27 00:15:32 2018 -0500

    fixed insert element bug

commit 4cbb8e0eb743b24415553a4ed16309abd7e28c43
Merge: d4e8eb8 99ebebe
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:57:39 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit d4e8eb805720d9ceb37bc9043252d91ecd116101
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:57:27 2018 -0500

    fix sast and syntax errors in semant

commit 99ebebe154628949170180f1a76b8bc67c7b8359
Merge: 229d460 6d07908
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 26 23:44:57 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 229d460165971b89e1847f7bff403134c2810a40
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 26 23:44:46 2018 -0500

    added functionalities for boolean arrays

commit 6d0790819d2c896158748de69f4318d895080027
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:42:01 2018 -0500

    change ops in begin
```

```

commit af6e7fc1e711fee9e20c77304cf505c91688b90c
Merge: 8ef6f68 3b53883
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:40:39 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 8ef6f6802df2f9fc1a50e1653f24f6988b9d10a1
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:40:16 2018 -0500

    changed increment and decrement

commit 3b5388318e10edecd870d7f8e65cd084d15d5451
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 23:39:35 2018 -0500

    add regex file

commit 2035771724a9a1e2b24aa779ca033a5d13b1e3ef
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 23:35:50 2018 -0500

    update sast with new binops and uops

commit 5b7c825eea0712f42640b34b2b7239d6df7ed951
Merge: e36fff6 67f1ac7
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 23:30:26 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e36fff6b63227e1c1d7f8164df1ed0a4822f9346
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 23:29:46 2018 -0500

    fix wrong binops and uops

commit 67f1ac7e1ea0e34c6e4652a2226df714811429f5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:27:09 2018 -0500

    change ops to expr

commit cee23e85f62917f59d6da276720a5e93cb273157
Merge: 92a67e4 13c980e
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:22:46 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 92a67e43fe1a7992fd9e5d3d6377c99a39fdc3bd
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:22:29 2018 -0500

```

```
    change strcat to expression

commit 13c980e58592d38d97176ea62d26fdc0e0a14694
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 26 23:22:00 2018 -0500

    remove builder from string format

commit bdeddd5eb0ee770de688844ac623691db6b6811c
Merge: 51b4ab8 028982c
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 26 23:19:14 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 51b4ab8f17cc2e0916f6b8a2f215d260b74e0533
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 26 23:18:49 2018 -0500

    add binop and unop to loop and end

commit 028982c7588b7cb6582c47571d7aadbfc4eaad36
Merge: cef698a 7e28721
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:16:22 2018 -0500

    xMerge branch 'master' of https://github.com/soybean/PLT-f18

commit cef698a20c92173db7d513fe2693cbbc9c2f14d1
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 23:16:03 2018 -0500

    cleaned up code and let $ not be called in begin

commit 7e2872130fff19d1bec91b44ea8988c71e538f31
Merge: fd37e12 e7c8bb7
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 26 23:09:46 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit fd37e12c619d452bda4bfbef7783fc9736c45df6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 26 23:09:36 2018 -0500

    added functionalities for string arrays

commit e7c8bb7572117707651b6eed5ea8e8e0d25a63b2
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 23:00:47 2018 -0500

    add gitignore

commit 9b5d56256965b95dc8b9e1467ff7c26240e41d48
Merge: e2f0bd0 40c3b3f
Author: Melanie Sawyer <melaniensawyer@gmail.com>
```

```
Date:   Mon Nov 26 22:58:12 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e2f0bd0424fef306ed10297adc8ff5f6a2e81a38
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 22:57:39 2018 -0500

    get $ working

commit 40c3b3f1bf95f3b5f9ed58dc726235eb77c75b82
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 26 20:54:48 2018 -0500

    add for, if, while to codegen

commit 29cea60bd717f44ffa67451963179ca6ac55d3ab
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 19:16:23 2018 -0500

    cleaned up deref

commit 07ab23d6fe31c8c87492e11576984b93abe7a7b0
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 16:14:58 2018 -0500

    fix return types for insert and delete

commit a7625fcd272c37211411f497a1287c89fc631fe6
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 00:25:07 2018 -0500

    update access in loop

commit 4cfe18b560f6735e9066576ff402b172140fe643
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 26 00:24:16 2018 -0500

    rewrite $

commit a52a4fb184e92e38aa7f51af7ef40973eff7988e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 00:22:21 2018 -0500

    add parser

commit 4db4fb665a52b37e02baed1259843f398788181c
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 26 00:16:58 2018 -0500

    add access as expr

commit acd6a9d07e865368d4a8b0ec4521a2f407984f11
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 23:29:55 2018 -0500

```

```
    fixed

commit 33a8970b2c61da67339950b01317d957f69f9f28
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 23:27:48 2018 -0500

    add access as expr

commit 2c5834de7c398a928c7689acb16747652c30bb73
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 22:01:32 2018 -0500

    add delete to loop

commit 877eb733243749302b75060069915ff42510c091
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 22:00:47 2018 -0500

    syntax error

commit 251f4c91d6bd66c292e0d67cbb5c3a8a4fe81f9a
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:59:22 2018 -0500

    prelim delete

commit bd25e6d88d236f7ea8edfd3dd252fbafbb4267f4
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:55:34 2018 -0500

    temp ty

commit a82a11ed48b80b0066d9988199b11a943a171e25
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:53:53 2018 -0500

    clean up

commit a3dc9504519890f85a10c8d3d7995f708c5f504d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:50:59 2018 -0500

    cleaning up unused arguments

commit 0ec0d495d7fd0770dd42bd08ad0cf4274bedf7ad
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:48:02 2018 -0500

    added insert to loop

commit 1e5cca68f3b88f1e00a46a0d26ce1ccc3938074d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:46:43 2018 -0500

    fixed insert

```

```
commit 4601946dfaa747299f2c49f98eb4e9dd2c83efc5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:33:54 2018 -0500

    wrote insert

commit 95ee1663dc8229aacdce9ef33834a6bee5a2d367
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:30:42 2018 -0500

    WIP insert

commit 54c14e4770c9108f2cf2e993c3f4e54da9f3f6f5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 21:20:11 2018 -0500

    remove access as unop

commit 3f37b75c7a9959f3bfaeffc322364f52e0ed96d7
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 20:00:14 2018 -0500

    change arraylit in begin

commit b5035672401e7636a8f35807f671e78e42bc72f1
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 19:58:26 2018 -0500

    checking array of different types

commit fac88dbebde940a68f78844654c1a6a8ee2f5d50
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 18:09:26 2018 -0500

    update arraylit in begin

commit 0a737c43aa975c3f4171e21d7110c5f48580c043
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 18:06:04 2018 -0500

    trying to fix type checking in arraylit

commit 500f710f01a91fb3aac852e13e4485ad6658bce6
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Nov 25 11:46:32 2018 -0500

    add default values to rs, fs

commit bb84ac3aa51228fd8284859f15c0b7e2a4b8be8e
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Nov 25 09:02:04 2018 -0500

    add rs, fs assign to codegen

commit 71f407d9b3db81b2f51a3cb59efe60ceefb04703
Merge: 90806df 015fe16
```

```
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Nov 25 08:59:31 2018 -0500

    Merge branch 'configbranch'

commit 015fe16a103d8463e522b49bd0af877df28f975b
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sun Nov 25 00:49:39 2018 -0500

    add rs, fsassign to codegen

commit 90806dfb43e2060c7b9423be47f2c3d6bf3b45cc
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 00:09:18 2018 -0500

    update arraylit

commit 3c07c6302d65a6f911d164a8851d4dcbf994bf3c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 25 00:08:04 2018 -0500

    fixed nested arrays type failure

commit f46bcc761466bacdd9410a02433bc95904961694
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 25 00:00:30 2018 -0500

    cleaned up code

commit 5d658eeb9536ab2a7c8584f8a73a350e7aa83879
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Nov 24 23:50:44 2018 -0500

    trying to add config, modified ast

commit 61c0d3909f96189aafd49ab51a0d7fbcc0b27ee3
Merge: e70b8ae 8bcfe77
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Nov 24 23:42:47 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e70b8ae2824e1541fc32037515455de693498808
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Nov 24 23:42:36 2018 -0500

    fixed some array tests

commit 8bcfe779da04070903e108817baa3fe1f088964d
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 23 21:23:27 2018 -0500

    remove Rgx from expr

commit 60b8e76d6364ac9e703c0fece08ddb156eece955
Author: Ashley An <ashley.an@columbia.edu>
```

```
Date:   Fri Nov 23 17:15:55 2018 -0500

    edited array implementation to be in one file and added array built-in
        functions

commit fda1ffb2da8c23cf3647359cac286b7689f0dbbb
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 23 14:16:05 2018 -0500

    created our own array data structure in C

commit adcbd583d6e0f3764e0526a5f88a14ef7a161f98
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 23 10:47:34 2018 -0500

    test script remove input.txt command arg

commit ef75916d79abb112a290fd958e081fb4aa805a0b
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 23 10:46:32 2018 -0500

    remove -f and input file name from test script

commit 7c0b36c4bd109d9c301df514caea2dad2870c181
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 23 09:33:06 2018 -0500

    added test that checks if functions can call other user-defined functions

commit e8c3290dcdb9fbd3ac11d2932cfcbad5bb2f5ea2
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Nov 22 23:10:17 2018 -0500

    add assign to codegen

commit f8e0585f51894f820f41bddf709e8409beda470d
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Nov 22 10:17:01 2018 -0500

    refactor loop and end blocks

commit 36345bd405c6fab97cc8cf9ed0d7d28057a6ba5a
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Nov 22 10:08:36 2018 -0500

    add locals to loop, end

commit accc673a8fd312ac0eb6175e4a2a3236895b42b0
Merge: c19f691 f7d11d0
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Nov 21 20:49:23 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit c19f6914d542ce6148549b6ed9c28d5fca41ff61
Author: Christine Hsu <christine.hhsu@gmail.com>
```

```
Date:   Wed Nov 21 20:49:03 2018 -0500

    codegen is now structured correctly

commit f7d11d064211c9b36338e1be7b79485c217f3e0c
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 21 17:12:29 2018 -0500

    add line parameter to loop function

commit 21145b8a9e6c6506d44ace534bb57c53712237c6
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Nov 21 16:56:49 2018 -0500

    add int_func test

commit dcf6db54884860998a0eb5495ccf514127112780
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Nov 21 15:51:08 2018 -0500

    begin block works for some cases, need to test more

commit c4bb4e6b30ca388f356981a71685fd8ae36cf92d
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Nov 21 09:42:41 2018 -0500

    begin block error

commit eee34d7c152c1b54695cc8039d4ba3b91048a240
Merge: 62ba5f7 bcddef4
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 21 12:52:21 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 62ba5f7b2d4e3da24baa461a100e13bea04e1273
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 21 12:51:32 2018 -0500

    fix string_to_int

commit bcddef42cabf8404cd48d10fee6f38b645bb4cb1
Merge: de0d273 a9b4529
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Nov 21 09:03:48 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit de0d2737997657a6173368c8f10712e9dc1c712b
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Wed Nov 21 09:03:30 2018 -0500

    restructured codegen to work with structure for loop/end

commit a9b4529fb96cc1f227c2a0ac746edf16f48c3983
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Wed Nov 21 00:01:08 2018 -0500

    added arrayderef to begin

commit 927b14fc571a49af6e38eb870896b950947a7683
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 20 23:59:51 2018 -0500

    recursive arrayderef for arrays

commit 42dc61a92f9edcb74b9db5f18ad48c5f87cd3941
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 18:01:42 2018 -0500

    remove -f flag and input arg

commit 422f39097ef34b66f587d0d9fd5955917d62b801
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 17:57:49 2018 -0500

    edit test

commit ae6ba275ea36d8151ec4e2286b7088f448e032aa
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 17:42:48 2018 -0500

    change test and input file

commit c02acc6dc5f93339806ef173128ccb09e69c54c5
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 17:24:54 2018 -0500

    add structure file and looping

commit a40aa57195a7fb9d3142769bdff0f6b294aa731e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 14:53:22 2018 -0500

    add structure file

commit 1ae1922f30a81dd5309e80f9d71195a428843b1d
Merge: f4cdf47 ea6cf60
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 14:12:46 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit f4cdf4796455636e3cdeb8d2762e4bc044d715e9
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 14:12:16 2018 -0500

    remove quotation marks from string

commit 6a35dfc0a1f6e91dab6d30e3ee5dbd48633491e5
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 20 00:43:28 2018 -0500
```

```

    get linking to work

commit ea6cf60447589a3eb97ff88e6cffe5f3d3773f8c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 23:18:50 2018 -0500

    adding arrayderef to begin

commit 364a0321a3a1c322a0b1a8a60225f888f537c3b9
Merge: 9aef102 a272ef9
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 23:17:13 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 9aef10242748a14a8d258f84de3b6178455f0a3b
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 23:16:32 2018 -0500

    array deref

commit a272ef9f1bd394e9f6c02195db7e9745e17e92bc
Merge: df3f56b bbc287d
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 22:11:30 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit df3f56b5624aed43bf9784f9e76ffc07f0537f0b
Merge: 1ae4dc1 52eb8c3
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 22:11:28 2018 -0500

    merge changes

commit 1ae4dc14d8e37023e9ecdac181831948c8e78931
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 22:05:46 2018 -0500

    edit codegen

commit 52eb8c3c453b0ac6a62e70f437c2ca0a03cdb862
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 22:04:32 2018 -0500

    put arraylit

commit 0669e9b2e4fddfcc775c369944a4dfbf162ea082
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 22:00:47 2018 -0500

    fixing arraylit

commit bbc287df6413611bb356ddce67c72ff4ddbab2b9
Merge: beccfe2 52eb8c3
```

```
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 21:57:29 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit beccfe25d71a9bafd80f471c1431c2681602b636
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 21:56:53 2018 -0500

    add string_to_int

commit 8aa00cc9bab5adc3ccfde057ce00f480aa732d21
Merge: 280f26f a2b6adf
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 19:34:05 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit a2b6adfe577acdcdbd9868adbaf089050f59b9cb
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 19:56:10 2018 -0500

    added index_of

commit 280f26f5e25606a95a3eb3d51958f09b9827a0ce
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 19:33:55 2018 -0500

    added more tests for contains and index_of

commit b85600cebfc15751f1ae96d2a6047a50ea093d6a
Merge: dbf0368 d0e41a7
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 19:25:26 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit d0e41a76aa328d8b72ed9917a2511e50014f48dc
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 19:47:59 2018 -0500

    put contains in both blocks

commit 4fa00b3c09b8366bb780d18d8bc6371e9a808396
Merge: b2dccdf 80cc2a5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 19:27:31 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit b2dccdf718604d7ecb386dd921ee95cdfb25576c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 19:27:17 2018 -0500

    wrote contains

```

```
commit dbf03688c0c93875a573e292aed6164ef6759c15
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 19:25:15 2018 -0500

    added tests for contains and index_of

commit 80cc2a57e2f02c31fa0bde5652b5908b366a00d6
Merge: ab2958a 256207d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 19:19:10 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit ab2958aef825bf5ac5f2fab74f85da56a4d3f6cf
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 19:17:49 2018 -0500

    fix codegen

commit 256207d235192d37f67f5cf27a0e981c59ae3023
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 19:15:38 2018 -0500

    added int_to_string test

commit 7253cbd1cec0891dfa7291a6cc0239a1aa4515a5
Merge: 1d31eb9 9449603
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 19:11:12 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 9449603b5a143c23cc3dd93553de39871a53ed0a
Merge: 3e145df 16cc561
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 18:49:00 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 3e145df8589246326b4cea5b705c634334bb6e27
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 18:48:49 2018 -0500

    edited dynamic array tests

commit 16cc561389bf20567dd81d3b9f7051abfcb94030
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:46:52 2018 -0500

    updated length in loop block

commit 2ea855215a41e07e69c322db2f4ae4ab8d9e92ba
Merge: 7aa8fe8 324732f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:45:26 2018 -0500

```

```
    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 7aa8fe816ab83bc2b91334d8fd21ba79bb7cb89e
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:44:59 2018 -0500

    restructured length and removed wrong tests

commit 1d31eb9745c99b7b1bd02634a29b6038931d480c
Merge: cea2575 324732f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 18:43:50 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 324732f884cbe260aa31d406ad8ac8e35dd376a5
Merge: e36cc34 d6fd3d3
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 18:43:09 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e36cc3425db595e4f98f694715feccc5c895a4fc
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 19 18:42:57 2018 -0500

    edited Makefile so it will remove convert.o when running make all

commit d6fd3d3239fadf6eb66471d11f8461e19e649c47
Merge: c2b4686 d2a2086
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:38:25 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit c2b4686172dba118014a1337749c97e467c0e79f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:37:49 2018 -0500

    fixed syntax error for enhanced for

commit 6643a33268453ef2b9376b01b547dc0339b36b69
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:33:01 2018 -0500

    update enhanced for in both places

commit 3891c3d763d6ece91bda2002edf947289cdbc882
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:31:14 2018 -0500

    fixed enhanced forloopsgit add semant.ml

commit 5c588fd5deaa8f042da19c71528d78c9ec834198
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:09:47 2018 -0500
```

```

    updating both blocks

commit cea2575b2b7d9579f505220e60d32e6ad27b410a
Merge: b0ecd2a e2533b2
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 18:07:58 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit b0ecd2a667bf2f848c6e599a62076156859f826a
Merge: 7863a29 27dbb12
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 18:07:51 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e2533b28d29e9f3770b6ee9db7034ce3dfbab16c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:07:40 2018 -0500

    commenting in arraylit

commit 7863a293d1da96a8f22469c90b8e2fdf89868dc4
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 19 18:07:40 2018 -0500

    add wildcard pattern matching to codegen

commit 27dbb12c1846248d4a72bd3ee1340b23977a0f1d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 18:06:59 2018 -0500

    fixing enhanced for

commit e2153498e790ffeae3f2e6ac392672dbca9c5d5e
Merge: e779dcd a1c4c38
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 16:50:18 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e779dcd009b60642c8087ebc3f960113ce8d6970
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 16:49:53 2018 -0500

    add convert.c

commit a1c4c3879724fa7c6fd66a490bd1227c3010f0a4
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 16:49:09 2018 -0500

    put length in both blocks

commit 25d551ab108e9b28a25f7d21fea86caf958495ea
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Mon Nov 19 16:47:06 2018 -0500

    got length to compile

commit e41551778d6b053f27c2838ded61eb7c91c9d394
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 16:15:46 2018 -0500

    trying to fix length

commit 86b36f38754881f36a03fffd32287db9eda916f1
Merge: a6fa325 eb9beba
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 12:36:12 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit a6fa32564c7cffff3758d89384b88e148aec2586
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 19 12:35:58 2018 -0500

    added tests

commit eb9beba1e21c618692edb500273b9874ab0023c3
Merge: ab77308 43b6cc5
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 11:51:26 2018 -0500

    add convert to makefile

commit ab7730829c88ceffd92cf671ddf8bd47927c703c
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 19 11:49:54 2018 -0500

    dis makefile

commit d2a20865aa4e42dc42c6f91823a638b2e8bbcdbb
Merge: 5ac9407 6643a33
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 18 23:10:28 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 5ac94078e7ebace57c129946fca78c979b94b3cb
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 18 23:10:13 2018 -0500

    created enhanced for loop fail test and 2D array assign fail test

commit 43b6cc5feec8bb2f9720459f39316575a508d951
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 18 22:43:58 2018 -0500

    added target to Makefile that allows for clearing .out and .err files

commit 3a04c7969c7ff62d53159c8dd92c38a7ab7a8049
```

```
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 18 22:13:05 2018 -0500

    added more array tests

commit 8bbf6937fc9850557a3894019cbf2f3bf825e8c3
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 18 22:02:13 2018 -0500

    added more void tests and edited regex tests

commit c77121ce40443cb700893e5ae764bf1d53402220
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 18 21:32:14 2018 -0500

    trying to fix arrays

commit b13a89016cd53ff8f9b16363aeab9aebeefcc183
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 18 20:53:46 2018 -0500

    commenting out builtin array functions

commit 5dcc76160524ebb691dfe31c5208531606b97fbd
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 18 20:46:57 2018 -0500

    rewrote length

commit c16a243505a63a4de3d84f6c7dc28658472bc471
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 18 20:43:07 2018 -0500

    fix empty array problem

commit 2783b1bd675563e9990c8446c2cffbead2f16ad6
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 18 20:40:26 2018 -0500

    fix length

commit 90c82407bfb5a6cb2be42f74f6ad8cd1acbf941f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 18 20:16:22 2018 -0500

    Add length

commit e426c4e275ecd5715ef90079b606918d3cb258c1
Merge: 5ff59c1 76bc3a1
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 16 15:44:47 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 1ef338731908d9b584660fd95c6df90d3087dab0
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Fri Nov 16 13:03:34 2018 -0500

    temporary stash

commit 76bc3a13658bee8fb2b2bdf9d14c45ed0555546c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 16 12:58:44 2018 -0500

    Make things the same

commit 5686d6f9ac273d92b83966fe45487f685d183212
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 16 12:53:41 2018 -0500

    fixed NF

commit c96045e7e68b742fd145b5bc6d1225be7856057b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 16 12:20:31 2018 -0500

    changed functions back to length in array tests

commit 32894a3b031fd0e6577f30807979756c4e3ab71e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 16 12:13:49 2018 -0500

    edited test script to create .out and .err files

commit c68129a5d42c2e78d90ec08b476bbb7be01a564b
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 16 12:13:43 2018 -0500

    fixing NF

commit ca602b25f565b83098fe65ef0cd967c52d3472b1
Merge: 70b995d 9ec7353
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 16 12:03:55 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 70b995d08132b10891959d909fa8179f5855b087
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 16 12:03:33 2018 -0500

    fix NF

commit 5ff59c1ac581fc323a95664b17fa019ded8f96c0
Merge: 1bada60 9ec7353
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 16 12:01:58 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 9ec7353d36f51e84337c7231f551f148abf60db2
Author: Ashley An <ashley.an@columbia.edu>
```

```
Date:   Fri Nov 16 11:01:41 2018 -0500

    added RS test

commit 378a27e7d88267ae433fb9c27b947f27c9485bbe
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 16 10:59:07 2018 -0500

    added tests for FS and RS

commit 9f31759d3d32a4d6414b7834068bbd9fed1684c2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 16 10:34:56 2018 -0500

    fixing errors in semant

commit b8526586f65973d66a9ce7c92c70aa903d334fac
Merge: ecf7b54 f4c3bfa
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 16 00:19:09 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit ecf7b545e8d4f2b729137318526a10d637d22be4
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 16 00:18:53 2018 -0500

    added more tests

commit f4c3bfaccbed47ee1fcafe7b8a522dd9fd174d0d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 23:57:46 2018 -0500

    fixed printing in tests

commit 25b174a94cdde9d5791ac1ee6c62e564dfb837d3
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 23:33:21 2018 -0500

    Update pass-fib.bawk

commit e29f8e642acf38c610a4e7df329ddb724be019a7
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 23:16:11 2018 -0500

    fixed enhanced for loop stuff

commit 2fb7fed00960d9a3f4fd99b93d6fbeb5694f8bbd
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 23:13:19 2018 -0500

    match sast to ast

commit 66be0df4a149d99238defba968099d8c7d7f18af
Merge: 2c06283 7f0bedd
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Thu Nov 15 23:10:12 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 2c062830e9e3a8b80116175c5aaa6914622e6675
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 23:09:54 2018 -0500

    idk if this works?

commit 7f0bedd355ff78a005ea973ef8bc202e7e8aa9fd
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 22:56:52 2018 -0500

    edited bawk.ml

commit 766d6af0e17c0687fc4daeb8db8d0b458ffd1dcf
Merge: 025891e 408f4dc
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 22:55:13 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 025891e3c733464f1cf9f0120f4071fecc6f3659
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 22:55:00 2018 -0500

    fixed increment and decrement

commit 408f4dc45a218eb35ec0103ab27e78ac0a1e8054
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:38:45 2018 -0500

    fixed line 79

commit 3bb199c92aabcd5f4d141b3ea881279d5fc10bb1
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:33:50 2018 -0500

    fixed locals

commit fc6812c15c2705c8021617bdda19823562aa48af
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:32:09 2018 -0500

    fix

commit 0575c1b3c74c728cc5fdfdbe7534ac5c4167fdd7
Merge: 2ac8735 d992cb3
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:29:35 2018 -0500

    Merge pull request #10 from soybean/victoriajyang-patch-1

    fix

```

```
commit d992cb3123b2349a88578bfcef6c0c6eb2bbcd8f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:26:44 2018 -0500

    fix

commit 2ac8735a6b51626b051080a43993cd3342fb9157
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:23:18 2018 -0500

    fixing syntax

commit 205292703f582576157734798782bb3f8c815202
Merge: 6550c4f a1b780d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:14:42 2018 -0500

    Merge pull request #9 from soybean/victoriajyang-patch-1

    testing

commit a1b780d1df44cd4d332f0f3fea80e185d330b79a
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 22:10:29 2018 -0500

    testing

commit 6550c4fad59316461a4ffbfafd56e6804ef54805
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 22:05:12 2018 -0500

    fixed regex tests

commit 23bb3dcf15252a492458767b2eaf75537b360334
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 21:41:33 2018 -0500

    Update pass-ops4.bawk

commit b0ee8e8acbcd73ff14451e28baeed60a307401c2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 20:56:29 2018 -0500

    match sast to ast

commit 8dae8d6f1547855ea2e37da6e5dff898cc0dc5fa
Merge: b8c3936 26942bc
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 20:54:04 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit b8c393687473b4fe03a797887f686d20e3b24a77
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 20:52:12 2018 -0500

```

```
    uncommented sast from bawk

commit 26942bc310ab5981a209e4760f6933fb932622ad
Merge: 58f7fb5 bbe7118
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 19:14:15 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 58f7fb553ae412c61e6a5c2b1f582e4a9a3fd502
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 19:14:05 2018 -0500

    fixed else stmts

commit bbe7118478793bff353962a87aa04399fe05ed57
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 19:11:37 2018 -0500

    Update pass-func3.bawk

commit 8d2d042b803a7ec3085f551ed09b8a6dab2259ae
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 19:03:11 2018 -0500

    Update pass-boolarr4.bawk

commit 6422d58e4db7715bafb555ab796da373c0caf1cf
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 18:16:18 2018 -0500

    fixed problems with order of statements, vars, functions in parser

commit ec0d36edbf23dafe955280451f7c1a01c171888c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 18:02:14 2018 -0500

    added to pretty printing

commit 4df85e256166881de1779bb65570c81087c54962
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Nov 15 00:22:16 2018 -0500

    creating blocks in codegen works

commit e298917446c00321d3b0f353ca2808d60fac0531
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 00:18:29 2018 -0500

    edited bawk.ml

commit aab137a7da6984c3512e7a0a5385291a727aa48a
Merge: 8f8f609 85a594e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 00:15:33 2018 -0500

```

```
    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 8f8f609bae1399daa2abb1ce43404ccc736991cd
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 15 00:15:22 2018 -0500

    fixed arraysvim parser.mly !

commit 85a594e16eac276fd18cb6993a7ed12092af383d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 00:11:44 2018 -0500

    fixed pretty printing syntax errors

commit c5438999627f31843145423bedfa1dc26d1c1b57
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 00:08:25 2018 -0500

    Putting pretty-print back in

commit bd8ad2a0ee18b072f8def8e67d42be2ec8d92741
Merge: ca1cbc2 bf80f27
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 00:05:10 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit ca1cbc29791fbf686d9af07098b53cba7b9aa15c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Thu Nov 15 00:04:53 2018 -0500

    added block checking

commit bf80f2769d6254d9add5c34a574c0d40db3b438f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 14 23:59:08 2018 -0500

    Cleaned up unnecessary comments

commit c6ccc613d3a76cc1f2b0e7e88e687341bb0803f9
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 23:16:49 2018 -0500

    added pretty printing for ast except for array types (like int[] or int[][])

commit 3c252d6abe0baa2068fe0194dbb9d3d5c622a05b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 21:35:00 2018 -0500

    clean up code

commit b3ef0f10f8c00eedfcbb8ee814e71e5a794349a4
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 21:30:26 2018 -0500

    cleaned up code
```

```

commit a4bc32564118eb6b7647300cb522256257b4e0e6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 21:25:19 2018 -0500

    cleaned up code

commit 96b7ae391d837cd548776cabfdb7623f378f0f06
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 21:22:04 2018 -0500

    more cleanup

commit 37a8853908832221ae6f3a34e18accf08b87e4be
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 21:20:57 2018 -0500

    cleaned up code

commit 7bb05e4368c3db3d4b4b58eb180141256961f555
Merge: 3619ed3 78e710a
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 21:13:40 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 3619ed31fd3de10730d1f11668535deb06ecc610
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 21:13:28 2018 -0500

    cleaned up code

commit 78e710a975d473172b0343240d289cfaa7e714eb
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 14 00:54:40 2018 -0500

    Reorder

commit bc8c8b9d544f8ca90b606e4df17b69f9435253d4
Merge: 48151fa 5d64af6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 00:53:58 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 48151fad979d4ccfb48f04194c6ef6b4a4f1aff6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Nov 14 00:53:46 2018 -0500

    created even more tests

commit 5d64af6b47d82660cfadbd070881d796034e106c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 14 00:44:06 2018 -0500

    Remove old comments
```

```

commit 4aace0fb9fea56fadd99a04ecc2d3ab29a6ec6a4
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 14 00:43:05 2018 -0500

    added type checking for globals/locals

commit 753f874934e48228c1137837d1bd879947e345c0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 23:31:32 2018 -0500

    modify test-hello.sh to run pass-helloworldloopend.bawk instead

commit 65c49c4fdbbc0055da6b3c963c6f2af091167358
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 23:23:00 2018 -0500

    adjusted spacing and indentation of test

commit 1bada60a830c2f713792bd646ebe25e8a222bfc2
Merge: 64dec61 153366a
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Tue Nov 13 23:22:13 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 153366a00eb10a55fd6aecc247ed590742af354d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 13 23:17:17 2018 -0500

    testing arraylist

commit fd99f2a97abac146908d5a45f3013c38254eeaf7
Merge: 1d14059 abe5513
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 13 23:17:15 2018 -0500

    Merge pull request #8 from soybean/christines-branch

    includes codegen that supports print in loop and end blocks

commit 1d14059f28810930aff0f1bf15406fefbd621e52
Merge: 5876d99 38a812f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 23:10:10 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 5876d99b84be061f206bd1a6332098be2018840b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 23:09:59 2018 -0500

    added more tests

commit 9c6c7a70c5b649450cbdd007408bc36e42eef938
Author: Ashley An <ashley.an@columbia.edu>
```

```
Date:   Tue Nov 13 23:00:03 2018 -0500

    added some tests

commit d55c391c67d346f513a9594dae824cb4d235f75d
Merge: 02c0fa5 df0476d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 22:58:37 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 02c0fa591eaff6b46678e3e5240da8d0fc631771
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 22:58:23 2018 -0500

    edited sast for return

commit 38a812f0dc72c61ddc82eb1455cce57b1808ab90
Merge: 088074c 3d3693b
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 13 22:43:35 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 3d3693b5f56fcf9b3a2a52ea216e09a0c33ae299
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 13 23:01:32 2018 -0500

    Added noexpr

commit 088074c5eab879551f426b9d48a8ce453dfc6856
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Tue Nov 13 22:41:39 2018 -0500

    add changes

commit df0476d8b4b696dc70db3ab84da6da77a51688a5
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 13 22:31:55 2018 -0500

    add make clean to test hello script

commit abe551358e00443aa69d79c8dee347eb65a8121d
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 13 22:05:11 2018 -0500

    add comments to codegen

commit 31626f01333aee55fe07285d7eaa93c3c90c7032
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 13 21:57:34 2018 -0500

    got print working in loop and end blocks

commit 94965e8c94cdb61bd52f13736b23194a321c6a72
Author: Ashley An <ashley.an@columbia.edu>
```

```
Date:   Tue Nov 13 18:39:37 2018 -0500

    added ability to write return ;

commit d6b9b7f8788449ce0a5d041a2738d2393ad69db8
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 18:23:48 2018 -0500

    modified some tests

commit ca0a491b05c584a4b93d2e117f78d0c02cc69cf2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 18:00:38 2018 -0500

    modified tests to be in end block rather than loop block so testing is easier

commit 7e323d8f0526451c9902f127c1df709aa2474bab
Merge: 87912b0 b0a11de
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 17:59:29 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 87912b0b2aa6ffe039972a7e27ef13fdec1b89fe
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Nov 13 17:59:16 2018 -0500

    modified and added some more tests

commit b0a11dea6ddfc2b0e51854db319c9f4782a9135f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 13 11:15:50 2018 -0500

    add make to test hello script

commit e17cc59e5528f200259f9a949163df186c9748c9
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 13 08:47:52 2018 -0500

    add test script for hello world

commit 2aadc7d77b591a56e521d19d98257e9f64291790
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Tue Nov 13 08:43:19 2018 -0500

    modified codegen

commit a119b50867ba2df843fe1c2928b53420d9d44219
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 23:44:47 2018 -0500

    Update sast to match ast

commit 66deb6be0baba1613fb1662b15f8be2ed69b1b0a
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 23:26:53 2018 -0500
```

```

    version of semant with NO ERRORS

commit fa7971fde838407306515204540377a8d46ec3c8
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 23:22:14 2018 -0500

    assign update

commit 5c007c1054d73c4276a6f4fadc2311afa76f4b44
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 23:12:57 2018 -0500

    Syntax error

commit 0d25831317f84d4bc5bb3e4ab9c382c037cdeaea
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 22:01:37 2018 -0500

    Commenting out pretty printing

commit 9f653986f5527465dc248b167254a61350067e84
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 21:51:34 2018 -0500

    assign is back to expr

commit e308c711b2643b7a4e6663079fe840e420597760
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 21:28:28 2018 -0500

    remove unnecessary test case

commit 35ce5f314bd62dd9799d9a86c2ecad2d01b6f87b
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 12 21:10:38 2018 -0500

    add print test for loop and end blocks

commit a3adcfdcb170fa6d186846441b8b291c17b87f89
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 12 19:05:47 2018 -0500

    testing branch commits

commit 64dec61a71730db43df3b87847312f50a9157c35
Merge: 1c57896 35ce5f3
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 02:07:47 2018 -0500

    Merge branch 'christines-branch' of https://github.com/soybean/PLT-f18

commit 1c57896471133e8f749bd11a6a8f370dd270f6e4
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 02:02:51 2018 -0500

```

```
    revert to id assign expr

commit 9577fec6f73757518e95a44ab81e6f05d06aba1f
Merge: a79b58e 953bd28
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 01:59:02 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 953bd28cd3c863f50bb93c21e33a5650b3f00267
Merge: 6e8afce e023701
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 12 21:04:44 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 6e8afced52e5307385d8b36de9a8a5a646e90afa
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 12 21:04:37 2018 -0500

    fixed bugs in sast

commit e0237012f5ccbc60892ed389d17ea16606f09c19
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 21:03:32 2018 -0500

    Changing assign back

commit c89af10bf58843070ffe782497fdd00048181273
Merge: cd42e82 8d6b98e
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 12 21:01:27 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit cd42e8242b180322e1b757d5894692448d1ad16f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 12 21:00:27 2018 -0500

    add fail assign test

commit 8d6b98e3655d69497fb646a1bcb33686655a4f8a
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 20:57:43 2018 -0500

    changed assign to ID

commit f298d7e6a27742681b1317404147e04720a7e8be
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 20:51:18 2018 -0500

    no keyword checking lol

commit cd13522c0f98bb88521c26a5d94de47bc4056c1b
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 20:46:41 2018 -0500
```

```

    reverting assign

commit 5670d61be3524cb2265bd4d51b1574efc0b161f2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 20:34:24 2018 -0500

    fix assign

commit ef1fab20e19460ec6b54e2cdb2f5037854b388a2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 20:19:11 2018 -0500

    added different types for contains and index_of

commit 99e45b20336c0a18c2f0fbefc624121a0a83c36d
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 20:16:54 2018 -0500

    testing contains overload

commit 8eb3715e79ffc8af7d8fa91bd0d395d997edbf42
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 14:16:13 2018 -0500

    helloworldbegin is supposed to be a fail test not a pass test

commit a79b58e92904ea1bd42e6b96713fbb45942f481d
Merge: 489d5b6 5b39b96
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 00:54:29 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit fbbe2f5a7e833c658726e7c089b7b1b7b1e7c353
Merge: 85aac4e 5b39b96
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 00:52:38 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 489d5b665236080c50611197fabdbf82c1df8c3d
Merge: 2d76716 9b0165c
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 00:51:04 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 85aac4e982e2e86acf88823298eb834002304d15
Merge: 25864d3 9b0165c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 00:50:28 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 5b39b96ca23b30e8a3311d77ff08c782eeb62875
```

```
Merge: b8e9582 9b0165c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 00:49:49 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 9b0165c2690eeed8467d74a09aad57410a97c094
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Nov 12 00:49:18 2018 -0500

    test Christine's email linking to git

commit 25864d338a2dc73000dd0bfaf7e9a7ab5ef39be4
Merge: d832c31 4823add
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 00:49:05 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit d832c3198444271d10b35bef804f46793222cb84
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 00:48:49 2018 -0500

    added basic test for dollar and basic test for nf

commit b8e95823e53c9804af2e15f2f4a4178794fabc48
Merge: 45e0f7f 4823add
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 00:48:17 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 45e0f7f252b5d9e7d00de4671da14f7c39c562bf
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 00:46:51 2018 -0500

    made change

commit 4823addc22d5cb7398f0eda4282d30105ead5a54
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Mon Nov 12 00:45:58 2018 -0500

    added more patterns to expr in codegen

commit 2d76716be73798c6ef9ca047c72fdce817f7fe38
Merge: a4d8f9b b89ef51
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 00:44:59 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 9632aaf623f1eac7423c3aa5b00e55702fba6a5b
Merge: b9006fb 0bb1ca2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 00:44:42 2018 -0500

```

```
    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit b89ef51279ab479be8cfd998cb788835f58d508b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 00:44:35 2018 -0500

    added makefile documentation

commit a4d8f9bfa036b6797ab8862f98f4cae12e2729bb
Merge: ea0b8ce 0bb1ca2
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 00:43:38 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 0bb1ca21bd9e884e1b7eef5bb9c9cf9b974da0d0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 00:42:51 2018 -0500

    edited Makefile

commit b9006fb31c1039ef81f65160b51323161a77c664
Merge: f2ab387 78688ab
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 00:28:19 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 78688abd8efa39651bba19006e17e0334c1232e4
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Nov 12 00:25:54 2018 -0500

    created complete Makefile

commit f9c13f58a51dcbb2cca56fde045482dfedbe8745
Merge: 49f02b5 43448eb
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Mon Nov 12 00:14:46 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 43448ebd6b6932ed747f3951bc2d9f406b7e9625
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Nov 12 00:13:17 2018 -0500

    add dollar and NF to scanner

commit f2ab3876a864a492e6baee9dcd15bed35c66ef86
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Nov 12 00:11:36 2018 -0500

    fix

commit b030197e55f4aa80f155c0a4e856acb694935002
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 23:47:35 2018 -0500
```

```

    fixed syntax error

commit 4b2cd8e7690b0f51eddec30b304c3b004689186c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 23:41:35 2018 -0500

    Revert

commit fd481f0007957ce2c029604077d08f748231690b
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 23:39:14 2018 -0500

    Update sast.ml

commit 49f02b58b96c4b539d892ad2e009a9d3e790553d
Merge: 9c4925e 8683378
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 23:33:48 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 8683378c472b8e52e81049472acbd9ec029c49fe
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 11 23:32:15 2018 -0500

    Note:  is supposed to be Ashley An (error from using git in VM)

commit 9c4925eff397559053e0c3601e053f2d987aaace
Merge: 8a10797 fb465df
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 23:31:07 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit fb465df3a1c0acec00ab87185534fc25d647dd56
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 11 23:27:38 2018 -0500

    removed line to test because name Alfred Aho is coming up instead of name
        Ashley An

commit 8a107976740903bc0e4c41d1acb84967a2b40ebf
Merge: f2e8269 f33ef97
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 23:23:06 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit f33ef977dafee702a1b2e49c118a67872b13d141
Author: Alfred Aho <Ashley An>
Date:   Sun Nov 11 23:22:54 2018 -0500

    added line to input.txt

commit f2e82699803e21cdfd0a2571576d0885e80ea12b
```

```
Merge: 092c929 dd3b216
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 23:20:09 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit dd3b216d8eebaa23898524bd751d94cce808402f
Merge: 2a47441 2f248fe
Author: Alfred Aho <Ashley An>
Date:   Sun Nov 11 23:19:13 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 2a47441321dd6b661b3b8664ae79a8f7222abf9b
Author: Alfred Aho <Ashley An>
Date:   Sun Nov 11 23:18:52 2018 -0500

    created tests for nested arrays

commit 2f248fe44ae9df36847df6174c198b556cbc70e5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 23:17:33 2018 -0500

    trying new builtin decls

commit 092c929d521665955d5a39fb6394da77b9a71727
Merge: 5f45972 609c8ab
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 23:04:44 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 609c8ab563fab9534b77688540bd26f0718f2509
Merge: d3953b0 61229b5
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 11 23:03:14 2018 -0500

    Merge pull request #7 from soybean/ashleys-branch

    Ashleys branch

commit 61229b58866cbb02d0d62ea2ae2902278291554a
Merge: ec163db d3953b0
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Nov 11 23:02:54 2018 -0500

    Merge branch 'master' into ashleys-branch

commit ec163dbb51048dbbe0082acf980e330f38595bfb
Author: Alfred Aho <Ashley An>
Date:   Sun Nov 11 23:00:53 2018 -0500

    edited bawk.ml

commit 3701ce5db31de8deea874649bac8c1fd6417d353
Merge: 6d24f20 7383211
```

```
Author: Alfred Aho <Ashley An>
Date:   Sun Nov 11 22:58:55 2018 -0500

    merge

commit 6d24f20476ce564784d5db2bf9f733ffef7a2ece
Author: Alfred Aho <Ashley An>
Date:   Sun Nov 11 22:56:03 2018 -0500

    implemented nested arrays

commit d3953b090e42a46b58e338711dbc50fa13fd8023
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 22:47:33 2018 -0500

    Remove map

commit 2eafa3866b6bcd81826f4530c74685cf50de70e1
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 22:46:59 2018 -0500

    Removed maps

commit b3629016630d5f0f6a0d5904fb276005ee69c13c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 22:46:09 2018 -0500

    Update expr

commit 03d76710bd4c446636df4023cba5ed3f306492b9
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 22:44:36 2018 -0500

    Removed map

commit ea0b8ce84d52c80b1cd042b7b9330d182480695d
Merge: 66ec1a6 d2bb3f0
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Nov 11 22:36:58 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 66ec1a606a1409daeba7303f54d44486136189c1
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Nov 11 22:36:05 2018 -0500

    pass in input file to translate

commit d2bb3f07a73ce2b4e2304ede3578f69d20db8c06
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 22:35:43 2018 -0500

    fixed mistake

commit 956a8f12af38327714f204924bed00bda32ea2df
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Sun Nov 11 22:28:51 2018 -0500

    added begin test

commit 4a0092711d8bfb9448db6c73357954c1820f263f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 22:27:53 2018 -0500

    Added loop test.

commit ad18ed964981f6682e8c49a516eed5f457b03ea2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sun Nov 11 22:24:24 2018 -0500

    Added keywords as "built in functions"

commit 5f45972757e4a29c8c262eb62aadd4448bd19c2a
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 22:22:27 2018 -0500

    first attempt in many to organize codegen for blocks

commit 27ea0cc7b475d85080cfe7c09b9340b7f3702043
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Nov 11 22:03:25 2018 -0500

    add stuff

commit 7383211be99af5c189b46f4bb96ee9e92c09b075
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 15:41:08 2018 -0500

    cleaned codegen

commit 99010b5d63b1e4ee66a608d12d85b5453cfbfacf
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 15:30:27 2018 -0500

    fixed strings and strlits for codegen

commit 48f32e15a673a9b7578defae6b1ea554bd49ad95
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 09:50:33 2018 -0500

    fixed stmt list in codegen

commit d3f58f97f2e32f6191cd0119bd5e0419acbc8bb7
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Nov 11 02:04:50 2018 -0500

    add input file command line arg and makefile

commit 98a096b34514ed38405c39b9a5a1ba03b73a9ae5
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Nov 11 01:58:56 2018 -0500

```

```
    add stupid hello world

commit 531f74c53d254715734622597c6e7981b9eb53d9
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sun Nov 11 01:14:23 2018 -0500

    codegen produces LLVM IR, doesn't call function in block

commit 70fa1656e0e28696a390ac61497e90263700eb84
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sat Nov 10 22:35:43 2018 -0500

    added main function to codegen

commit 4b1cc1325683dcfed6f7b7092c644852ee4ff969
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sat Nov 10 20:27:12 2018 -0500

    broken codegen.ml

commit 3a6726ecbde01e82138d73ed5b928fbe006d0ba5
Merge: 3eea685 5d8f2f6
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sat Nov 10 20:09:58 2018 -0500

    fix merge conflicts

commit 3eea685b83c9dda9f51cda87852eba7515a30f21
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sat Nov 10 19:42:34 2018 -0500

    re-add codegen.ml

commit 5d8f2f613dd9a37c423cea13e3bb8e71e9486266
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Nov 10 19:39:47 2018 -0500

    comment out string_of_sprogram

commit 74c81549ef5aa15ff7022d4395a7d30d6592df8e
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Nov 10 19:37:21 2018 -0500

    add flags to bawk.ml

commit 8f7eb0e8cca0d45efac2b38154cb46874087eab4
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Sat Nov 10 19:36:35 2018 -0500

    add bawk.ml and sast.ml

commit 547a0e4ab7386b2e9785e6d98cff95b9cf59a377
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Nov 10 19:28:35 2018 -0500

    codegen builds but is definitely very wong
```

```

commit 1f035b70cc5992c26987c8a611f627e6efaf953f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sat Nov 10 18:31:08 2018 -0500

    Remove float

commit fa4efaf1d07499bf508a36b9ae0846bf84c9d0a2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sat Nov 10 17:32:33 2018 -0500

    Cleaned up comments

commit 0cede7240d81c80f2d70294f6e1b370f265425ea
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sat Nov 10 17:07:02 2018 -0500

    Added enhanced for

commit 692cc1e081c6f75d8afcd709920a756666b57691
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sat Nov 10 16:55:40 2018 -0500

    Added exprs

commit c73b47d29c17a4a7d550bd6e9d439ab61d3dabf8
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sat Nov 10 16:50:46 2018 -0500

    Update binops

commit 99a30cea1789b9b9f34d63a16f84f78ee8dba3da
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Sat Nov 10 16:23:46 2018 -0500

    Added our unary operators

commit 8c97e454de28e2d2817d2cf2ca07e06a2f148065
Merge: 3d37bfa 92529fa
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Nov 10 16:21:16 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 3d37bfaab8a2091e759e79b374b9ac2786791283
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Nov 10 16:21:12 2018 -0500

    got rid of ability to make a scope

commit 92529fa2f7850a48a1404a285239cb94aa90b624
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Nov 10 16:15:52 2018 -0500

    updated codegen, does not work still

```

```
commit 94c895ac0b9bdc9354e502958926ff639ac823c0
Merge: 35b1420 a67b6b6
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Nov 10 16:12:42 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 35b14209de1ee703d643464234ea2c639ff27db2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sat Nov 10 16:12:38 2018 -0500

    array get element and array assign element should not be unary operators

commit a67b6b69e53605fed8baf285ceaf8b755b699d33
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Nov 10 14:20:53 2018 -0500

    preliminary codegen for hello world

commit c60b174d9636cf28c89cf6fb27a95338e65db1f3
Merge: 3345110 07b3f87
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 21:13:20 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 334511083497d28dea61bc06d90bc7676de5e6f7
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 21:13:14 2018 -0500

    created hello world test

commit 07b3f87e993f7a21dc7d253fe0e42f78c1d99afa
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 21:12:11 2018 -0500

    change HI to PASS in sast

commit 6efca2f506316217601214d88f9cb7b1aae3d327
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 20:33:06 2018 -0500

    update SAST to match current AST

commit 567055834f26082d4397404a8d9b7381c5259c5f
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 20:25:42 2018 -0500

    update bawk.ml

commit 920b29f317197776e821bad080dcc9b62df184ae
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 20:23:57 2018 -0500

    update semant.ml

```

```
commit e4f6316e6df55fc2ac7f5bb4bd6f484f83f2cf9a
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 20:16:04 2018 -0500

    changed HI to PASS

commit 6b065c770cf80974628437b98c1a2e2c1f9667a1
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 20:14:09 2018 -0500

    added new line for readability for tests

commit 5609e9ccc6fb1b284a3a33c959ee9cb822c7248d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 20:12:54 2018 -0500

    fixed regex reading

commit 75f468177f87093b44b59a8b5815ac3de22535b3
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 20:08:20 2018 -0500

    got rid of duplicates

commit ba00d24d31c2b24aebde39347dc27a067ae48712
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 19:53:02 2018 -0500

    added more map tests

commit af35a6f841f7b5bf14e7fd23c0828461b707453c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 19:45:54 2018 -0500

    fixed config and enhanced for loop

commit f5687785d5308ee7ffc031e67d186043f0611600
Merge: 6f1796c ddc593d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 19:23:23 2018 -0500

    merged

commit 6f1796c6c372fbcf2975ea2eab9a7cc3b7e9ab7f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 19:22:20 2018 -0500

    fixed arrays

commit ddc593dded8f5c836d4ada928cab8ae7106a20ed
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 9 19:17:09 2018 -0500

    fix empty block problem

commit 14e93db6d2d084260a882544516cac6adefc76d5
```

```
Merge: 2f3b529 71fd9ce
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 9 19:09:22 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 2f3b5290034646b25f7b943eb3d2690da01fbfa7
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 9 19:09:13 2018 -0500

    get rid of reduce reduce errors

commit 71fd9ce8a8ef08fac6bf42932bba1c89a87e7189
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 18:59:52 2018 -0500

    Add rgx

commit c7f58d544242dbefe206151b4448826c648269aa
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 9 17:50:05 2018 -0500

    fix empty config block

commit 645bed233204dc74cc4b459a0b57363c36fd08c5
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 9 17:38:01 2018 -0500

    add empty map

commit 5ea2f5f56d82d3a98da0360d7910270a8f62426a
Merge: a9d61cb e4b9685
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 9 16:18:59 2018 -0500

    fix merge conflicts

commit a9d61cbe8cbd6e8e97350658c963931ef6438259
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Nov 9 16:15:02 2018 -0500

    empty map

commit e4b96851cc4fbd4cbf4ec03e22f249f0e913fb3d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 15:54:29 2018 -0500

    edited map test

commit 848823b3a1a3fd17e2d9bec46f1923df43d63d6a
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 15:21:19 2018 -0500

    modified map1 test

commit 411d64c472613072d911a12713165c7ec4f52a94
```

```
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 14:52:15 2018 -0500

    Move test-script

commit 087dc405dc8131cabd00b3220cec9f26381a317b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 14:51:05 2018 -0500

    fixed script for running all the tests

commit 21eed276a2d9221a8bf58fdc343e0b6098985f07
Merge: 24edf97 4cbdebd
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 14:43:07 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 24edf974eccb06af5880e0cfeb560d31c9deb450
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 14:43:00 2018 -0500

    changed names for array get and assign to be clearer

commit 4cbdebd927e44bba378210316d82714e560f678a
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 14:42:46 2018 -0500

    Changed builtin functions for new array definition

commit bae6a54ad6e9f0a96fa253f6d3f273da35bdb5d1
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 14:40:29 2018 -0500

    fixed arrays and added array tests

commit 90006336b26182f5c9e2046ad6618b158b7ac3d9
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 14:30:29 2018 -0500

    Create test-script.sh

commit 8ff0abe533e7cd8ebd37428b0f3c4c1c7c210acd
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 14:18:51 2018 -0500

    Added builtin keyword list

commit ffb82d418f3c27f80b3863771492f0b0f467a29d
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 12:52:18 2018 -0500

    fixed config

commit 37e875cbfd788fa31ffb792150c8102ac0738036
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Fri Nov 9 12:41:01 2018 -0500

    Removed extra in

commit 90de74fdb341acaa87c5ff0b9670742557660ee4
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 12:39:45 2018 -0500

    Returned stuff that I deleted

commit 927995ff903de01635f68b6e215df26288c533b5
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 12:27:07 2018 -0500

    Fixed int to string declaration

commit 189cf159dd4973b746af8c43aab3ae1b9972f630
Merge: 326d74b 45e5ad3
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 12:21:05 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 326d74becadf38c8738e9de7990fefa6fdb7b6f2
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 12:20:56 2018 -0500

    modified regex tests to have rgx_to_string

commit 45e5ad3b5da0e8fe383c20b8e2c3af230768f6ba
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 12:19:07 2018 -0500

    Added to string functions

commit 3729c4f765e41df6ee523de31657e94deac81c56
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 12:16:05 2018 -0500

    modified empty map in test

commit c4cca1ebf4321c6dc5b713c768204b3f25b07dc3
Merge: 0f80933 f8f42c5
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 12:09:06 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 0f80933c83115cf3abe0bfe8c45681ba33447812
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 12:08:52 2018 -0500

    modified files and added more tests

commit f8f42c533245b8b467f55afdc437dd40dd9f7c79
Author: Victoria Yang <victoria.j.yang@gmail.com>
```

```
Date:   Fri Nov 9 11:10:12 2018 -0500

    Remove floats from semant

commit a214584393a6fe70110dbfe1dcf72545d23c1db4
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 10:56:37 2018 -0500

    Update built-in functions

commit 3acb96f4067101aebc59b3f6608cc70088ac5404
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 10:46:10 2018 -0500

    Changed symbols back

commit 95ca0091c052f67557b2aac3f07631bab6b155e2
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 10:44:40 2018 -0500

    Updated typing

commit a1952cb36a6e7a19d4cf251b076792eab26a5d25
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 10:37:59 2018 -0500

    Added println

commit 7448f211c6c371dd75bdaf4a16aa90af64741b8c
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Fri Nov 9 10:33:28 2018 -0500

    Added built-in function declarations

commit f50196261d337487d108916625a44fa631250fba
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Fri Nov 9 08:31:25 2018 -0500

    add RgxLiteral

commit 638f2cae05f9b192ddae2a08407bd40cf8879189
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 23:02:07 2018 -0800

    Update ast.ml

commit 029a8e121a48b07c9d1e279cf79e32d51bd46247
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 23:01:41 2018 -0800

    Update ast.ml

commit 056066f6b7510e608ed5f31a75c51b6d67692a07
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 01:55:46 2018 -0500

```

```
    fixed decrement and increment tests

commit 3fa2a4a8dc64b68ff93784c4e931782e26d77f1a
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 01:39:21 2018 -0500

    Update scanner.mll

commit aa62f06e60713e1259f13425f4438cb17a4f102b
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 01:36:23 2018 -0500

    add stringliteral to ast

commit 9e5735806791baffea1f5879a544803ebfd1cdf5
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 01:35:14 2018 -0500

    added StringLiteral to scanner, pass_string3 fails

commit 02cd342ed76880da62a83d8c8daf4b7d56d82503
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Fri Nov 9 01:34:43 2018 -0500

    added StringLiteral to parser

commit 1480906f5eeee35d4d4754c7282149cc6ec1fbf8
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Nov 9 01:23:02 2018 -0500

    changed map to be defined with LT and GT tokens instead

commit 89a5d2bdb74bc134b389fbe30b899e1d88a1eb83
Author: Christine Hsu <christinehsu@Sam-Smith5s-MacBook-Air.local>
Date:   Fri Nov 9 00:30:27 2018 -0500

    fixed parser

commit 25adeb93a6f27c7e040aac36ab6224f6bcf699f9
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 22:35:56 2018 -0500

    added typ for function declarations

commit 12d6d50f5e07e5cb9c56d690674d74161060e37c
Merge: a8f1e6c 63c4015
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 22:12:16 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit a8f1e6c36c2c67dfcd1ed7ebaa7ba8cd24e8fc5c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 22:12:09 2018 -0500

    fixed print function in tests
```

```

commit 63c4015c522346ab80e5814c41399f3196c4c40e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 14:25:49 2018 -0800

    Update pass-add1.bawk

commit a7279228d66144979556e69f42f75a3a48f00a50
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 17:25:08 2018 -0500

    fixed config to be at bottom

commit ae3e5bc35e1b212ad1102c1a9b7747715d0a1f86
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Nov 8 17:18:37 2018 -0500

    added some tests

commit e904645fdf1f5c7f4eb73f83a1ac41c2371b038e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Nov 8 16:49:46 2018 -0500

    update empty map syntax

commit f7953cff7db70ad73d0777613bb6ea2ade68d802
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 7 22:12:30 2018 -0500

    Make sure rgx can't be in list

commit 051add2399c792e0344c4ccd2e975f6630e28570
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 7 19:50:54 2018 -0500

    Removed built-in functions we don't have

commit e35ad95412e5c73270b57ff11d1da34b3ac66077
Merge: 72bc22b 6686706
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 19:44:40 2018 -0500

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit 72bc22b64be9d1e99a2c2bda2b3a3a9e48918d30
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 19:44:28 2018 -0500

    add semant

commit 668670621f1ac9051063cc9ccd53b14c915fd329
Merge: 7e3c052 ea3518f
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 7 19:43:08 2018 -0500

    Merge pull request #5 from soybean/victorias-branch

    Victorias branch

commit ea3518fe2e9f185791a925f5aa40efdfb222bf10
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 7 19:42:33 2018 -0500

    Changed sast.ml to ast.ml

commit 7e3c0521047ca81c71a472f5192832dbc6747731
Merge: d221afe defe120
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 17:41:49 2018 -0700

    Merge pull request #4 from soybean/dev

    fix config block

commit defe1206605170f5a8fd2e1c3d1f0c11e7c5d293
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 19:41:19 2018 -0500

    fix config block

commit a9f6fcc00885569ba0329a9c23cd954822609b9e
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 7 19:34:39 2018 -0500

    Made copy of ast.ml in sast.ml

commit 318c188474e2f2975eb216db5803234e6ad78760
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Wed Nov 7 19:34:03 2018 -0500

    Create sast.ml

commit d221afe84de6be9eb4624997be3617aab1d64c4c
Merge: e9a7549 d31a8e5
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 17:19:04 2018 -0700

    Merge pull request #3 from soybean/broken-stuff

    Broken stuff

commit d31a8e5b79bdca1fe96d49464b7faa8cfaeb9710
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 19:17:35 2018 -0500

    fix config

commit fffa080436930448dae3f2879b60ad2589d86f3b
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 19:15:58 2018 -0500

    add config
```

```

commit e9a7549fec7219d1693175c53633043f8215ab70
Merge: 1428eb7 c42d710
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 16:53:42 2018 -0700

    Merge pull request #2 from soybean/broken-stuff

    Broken stuff

commit c42d7100ef175d74060e9063eb434955de35e7ed
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 18:52:52 2018 -0500

    fix regex

commit a6db87815d2e36165ce09e1bbfc23f58d18c82c4
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 18:49:20 2018 -0500

    add for to scanner

commit 7aefdd502f21fa840a83d4a9c44570b96820b4e3
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 18:48:26 2018 -0500

    fix parser

commit f0d15ea60c0eb958ba684716741c1c7a7ab04103
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 16:56:37 2018 -0500

    fix increment

commit b7e93c8fe2a543b232c91e561c186223f3f459b8
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Nov 7 16:37:39 2018 -0500

    add updated ast

commit 1428eb77a092025b1a0827684316a53a577661fd
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 29 23:31:32 2018 -0400

    fix syntax error in scanner

commit 419d90cbbed96ea9cd5503e54b2b83099307e696
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 29 23:28:59 2018 -0400

    fix syntax error

commit 56d7bf6773534b63daa5f11670151ae2faccc9a9
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 29 22:50:10 2018 -0400

```

```
    broken stuff

commit 6c910adf0dd7e572c8c826568b7d0b58b966cc55
Merge: e8ce506 ad12dce
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Oct 28 21:22:12 2018 -0400

    Merge branch 'master' of https://github.com/soybean/PLT-f18

commit e8ce5060f582d2f4dff230cfc8394b78b1fba37f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Sun Oct 28 21:21:48 2018 -0400

    added bawk.ml and made few changes to ast.ml to fix it

commit ad12dce46a66dd4402b8b7909cb2cd36b176b278
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Sat Oct 27 23:52:54 2018 -0400

    Create semant.ml

commit 4ee56f480c55b82eff8da18abfe12460ab0e10e7
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Oct 23 00:28:44 2018 -0400

    alphabetized names in README.md to test committing from local machine

commit 6e5ba1d48288fa478b819eb91820a4eaa5c5a107
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Oct 12 14:08:55 2018 -0400

    add all necessary updates

commit c4e72613803f364405091e9666aa04ce2fb15645
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Oct 12 13:59:34 2018 -0400

    move statements to expressions

commit c956e7353c099dc4e60a10976ad0986f567e405e
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Oct 12 13:36:14 2018 -0400

    add no else

commit 7deb0ee95fedd33b9d28328d6d6a08c9df0fe940
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Oct 12 13:21:40 2018 -0400

    fix expression list

commit 5474c104216caf56dc22d32ce95a3219c0da9f4a
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Fri Oct 12 02:42:55 2018 -0400

    final updates
```

```

commit f560251bd80edf7bc786afeb2bbb54db9e4ce763
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Oct 12 01:23:07 2018 -0400

    Update scanner.mll

commit bc458f55c22b707f9993852342fb27e395f53013
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Oct 12 01:22:54 2018 -0400

    Update parser.mly

commit 14acae7e2c60f23c32c2b500f11d2c990488b46c
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Oct 12 01:22:39 2018 -0400

    Update ast.ml

commit 9dcfe0a7eeb6f54305f493e9f14b7ac7205a1d31
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Oct 12 01:09:57 2018 -0400

    Update scanner.mll

commit dd6e13d8ff721b902b0b812192d71b0b0aee2d2e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Oct 12 01:09:42 2018 -0400

    Update parser.mly

commit b3f02a42588ddbe2325bea57e7645de2e84ab277
Author: Ashley An <ashley.an@columbia.edu>
Date:   Fri Oct 12 01:09:23 2018 -0400

    Update ast.ml

commit 6227021988cf2bf0043fbd04231a9567f23eb83f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:53:22 2018 -0400

    Update scanner.mll

commit 8aa80ae37cf932e47f67fc8e15f2e04ae56c2045
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:53:06 2018 -0400

    Update parser.mly

commit d232fc90731e6cdecb993a6f3e25946125f77101
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:52:51 2018 -0400

    Update ast.ml

commit ef9eabab57a5e26cdf585fc65709613373d4ccf9
```

```
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:34:28 2018 -0400

    Update ast.ml

commit 6fcc764bd4c098b71f0f3d79dd4e371d0d6282d8
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:33:20 2018 -0400

    Update scanner.mll

commit a2c48c787c095f77d398fa0f5b78286091a0d510
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:32:55 2018 -0400

    Update parser.mly

commit ec37941a07868847fbb69239e5460816a90f936b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:32:35 2018 -0400

    Update ast.ml

commit 405379de1662bbcc7cbfe98a93263e8d45143bb4
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:21:29 2018 -0400

    Update ast.ml

commit 3b4699aa53619c5c50fe7954333f34289ab45305
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:20:58 2018 -0400

    Update parser.mly

commit 1efdda81904c619551f772c82b3235efddbda2ce
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:20:26 2018 -0400

    Update scanner.mll

commit 5a0d628139c3659c950325ec995bc2b7884b459e
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 23:05:58 2018 -0400

    Update parser.mly

commit 6dc5e3de6fb35c9f71af596a4c9579847a558f17
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 22:34:22 2018 -0400

    Update parser.mly

commit 778f19e9695ac2593bcd8770fc04e67dcf9d7353
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 22:33:16 2018 -0400
```

```

    Update parser.mly

commit b6ff7dd4a7f151664da50cf7595b833ad92c4c67
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 22:32:33 2018 -0400

    Update scanner.mll

commit ac06d5803e0f7d7dde4c216631b563b728fe7280
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 22:32:13 2018 -0400

    Update parser.mly

commit ea6db582f04c5d6f5f3303f57e09bbfa56c15496
Author: Ashley An <ashley.an@columbia.edu>
Date:   Thu Oct 11 22:31:49 2018 -0400

    Update ast.ml

commit b6ffb1301951ee0d6af3b305b0c81b812853edb4
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Oct 11 21:59:45 2018 -0400

    Update parser.mly

commit 41839f00950cd652639a453bbe599ac57ca76b19
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Oct 11 21:52:33 2018 -0400

    Update parser.mly

commit 01096c9692865c865c5cd47aa3bacc187c19c9e5
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Thu Oct 11 21:34:56 2018 -0400

    added declarations to parser

    CONFIG block is unfinished

commit 25c469e85917f22157358f1b8b6a4ff610bba72a
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Oct 11 00:31:08 2018 -0400

    Update parser.mly

commit 033a3e04ab694dc643ac5e2eddd6075e9cf85339
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Oct 10 23:49:06 2018 -0400

    Update scanner.mll

commit e0597158056e51f8f44eef1c04619b9aad2a8e3a
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Oct 10 23:07:45 2018 -0400
```

```

    Update parser.mly

commit 49445099dd53a452d6ed13c020f3271928567f5d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Wed Oct 10 23:07:22 2018 -0400

    Update scanner.mll

commit dfe5e4d36b8cdbe21270cae34283b747cfd2c879
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Oct 10 22:36:42 2018 -0400

    Update parser.mly

commit 2ff7381a0773c35c7f0916caed6dab72964594dc
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Oct 10 20:30:11 2018 -0400

    Update ast.ml

commit 1f1cf3218f5ddcd135aca061c8176fe22a3fee0f
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Oct 10 20:28:24 2018 -0400

    Update scanner.mll

commit f8de63a86a8180a45a8320a7d65137f5063ba0ea
Author: Ashley An <ashley.an@columbia.edu>
Date:   Wed Oct 10 20:26:41 2018 -0400

    Update scanner.mll

commit 12940233aa2decac34e4ecbb6d42c7b4fbf66c3b
Author: Ashley An <ashley.an@columbia.edu>
Date:   Tue Oct 9 00:00:53 2018 -0400

    Create parser.mly

commit 637b93adb12cf2ef4e253c6166d4497da4ee25b7
Author: Victoria Yang <victoria.j.yang@gmail.com>
Date:   Mon Oct 8 23:56:21 2018 -0400

    Create ast.mli

commit a3297512472614f648925a9b9845f566bd462fc7
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Oct 8 23:55:18 2018 -0400

    Update scanner.mll

commit 272e7ddfd0601db23cd7127660ca884ebddd42c6
Author: Christine Hsu <christine.hhsu@gmail.com>
Date:   Mon Oct 8 23:54:47 2018 -0400

    Update README.md
```

```

commit e7e68c5d8d8b8c01097dbac0843b4ab629991bdd
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 8 23:51:31 2018 -0400

    Update scanner.mll

commit 4b49aa26fae72a802ceb48153066de31973b106d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 8 23:41:25 2018 -0400

    Update scanner.mll

commit 17b6f846178d655ed2e90d13bde54fe05c9baec2
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 8 23:39:30 2018 -0400

    Update scanner.mll

commit 4c738cd59c36bafc02abaef235cfdf6ed0fd507f
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 8 23:28:10 2018 -0400

    Update scanner.mll

commit 7c2e939c377d1549e4f9a751e0cb3fbee8ec0348
Author: Ashley An <ashley.an@columbia.edu>
Date:   Mon Oct 8 23:18:59 2018 -0400

    Update scanner.mll

commit ed86b16ef8a864bcc2ace0ed66c0d12617ee166f
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 8 23:12:42 2018 -0400

    Update scanner.mll

commit e1f203b5a4e55602e0d65da45bc8be66692bac9d
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 8 23:12:31 2018 -0400

    Update scanner.mll

commit ea77a459ca113570f414227cb3fd92f8df437c68
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Mon Oct 8 23:00:52 2018 -0400

    initial lexer

commit 5e80a92fae1b7ac108808b401bd7df764f3bd082
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Sun Oct 7 04:20:01 2018 -0400

    Update README.md

commit 53d24f473d55634f05b1d4e49e121fd7021de4fa
```

```
Author: Melanie Sawyer <melaniensawyer@gmail.com>
Date:   Thu Sep 6 16:57:08 2018 -0400

    Initial commit
```

4.3 Development Process

Our development progress traced through the stages of compiler architecture. We started with the scanner, then the parser, and worked on the semantic checker and code generation in parallel. In an effort to start code generation early, we initially used the AST rather than the semantically checked SAST, and then switched back to the SAST once the semantic checker was complete.

During the beginning half of our project over the front-end stage of the compiler, our team met individually once a week and then on Fridays with our TA John. Our meetings with John were flexible and usually consisted of us asking him various questions. Each member of the team contributed evenly, and we maintained open and constant communication throughout the project.

4.4 Software Development Tools

Each member of our group used vim as a text editor. To build and run our project, we used the `numel` virtual machine provided by John. As a version control system, we used Github. We mostly stayed on the master branch, but often created branches to test volatile features.

4.5 Programming Style Guide

The following summarizes the style guides our group tried to abide to.

- All variables are snake case, all C function declarations are camel case.
- Tabs are equivalent to four spaces.

- Keep lines 80 characters long.
- All variable declarations should be at the top of the file. Global variable declarations in the C files should be kept to a minimum.
- The C files are split up by the category of their function, and additional functions should be kept in the proper files. (`convert.c` maintains all functions used for converting or altering data, `rgx.c` maintains our regex library, `structure.c` contains as little possible code and should only be code that is directly executed when the program is run.)
- Code should be commented as thoroughly as possible. Large chunks of OCaml should be preceded by a block explaining the function.

4.6 Timeline

Task	Deadline
September 19	Project Proposal Submitted
October 15	Language Reference Manual submitted
October 15	Finished scanner, parser, and AST
November 14	Hello World program compiled
December 2	Semantic checker finished
December 13	Code Generation finished
December 19	Project presentation and submission

4.7 Language Evolution

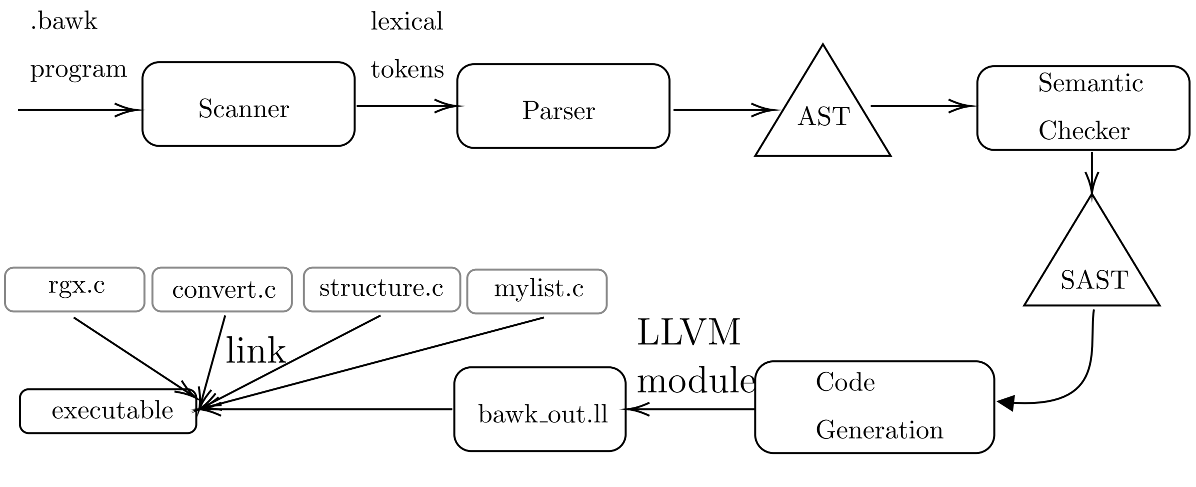
Overall the structure of bawk did not evolve much over the course of the project. We removed a couple minor features we had planned on implementing, namely enhanced for loops and maps (i.e. associative arrays). We chose to remove these features largely due to the fact that bawk contains normal for loops and nested arrays already, which achieve the same functionalities as enhanced for loops and maps would have. Aside from that we were able to implement every feature we set out to do.

One major change that happened during the implementation of our language arose during the semantic checker to SAST phase. This required changing the structure of a program by hoisting blocks into functions in the semantic checker. This made it easier to implement the `LOOP` and `END` blocks in IR code generation and allowed us to avoid writing duplicate code.

4.8 Testing Process

Throughout the development of bawk, we tested each new stage with unit tests. Then, we ran tests that were written for previous stages to make sure that they were still behaving as expected. More tests were added when it was still unclear as to what exactly was causing problems. Each of these tests targeted a specific part of our language, and having these tests that covered a wide range of our language features helped us pinpoint what exactly was working or needed to be fixed. More details about testing are included in the Test Plan section.

5 System Architecture



5.1 Scanner

The scanner performs lexical analysis on the input file, reading in the raw text and converting it to lexical tokens, which are in turn passed to the parser. The scanner also removes all whitespace (spaces, tabs, newlines) from the program and converts escape characters to single characters. Finally, the scanner removes quotation marks from string and regex literals, which prevents codegen from having to manually strip quotation marks from strings. The scanner will terminate execution of the program if unrecognized tokens are passed through, but syntax and grammar issues could still be present.

5.2 Parser

The parser reads in lexical tokens from the scanner and generates an abstract syntax tree (AST) based on the grammatical rules of our language. If an input .bawk file successfully passes through the parser, it is guaranteed to be syntactically correct. There could be semantic errors however, which are identified during the semantic checking and code generation stages.

5.3 Semantic Checker

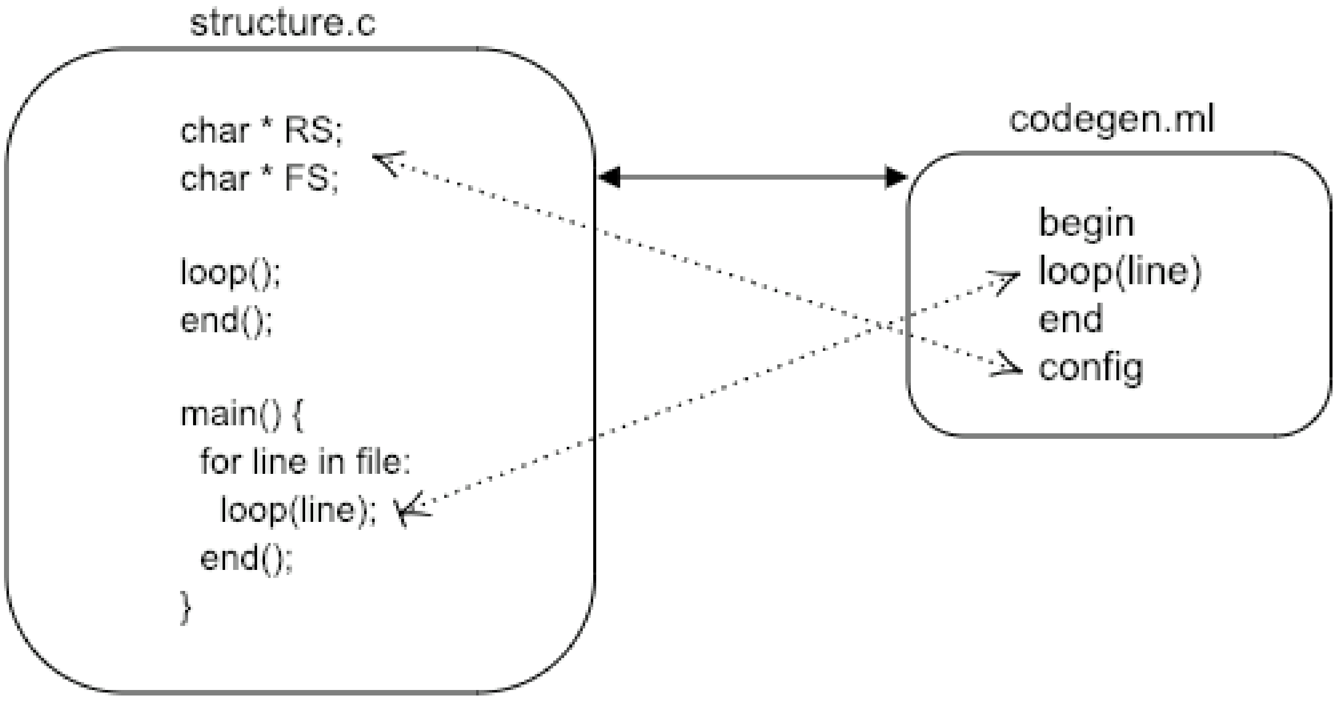
The semantic checker recursively traverses the AST and converts it into a semantically-checked abstract syntax tree (SAST). The semantic checker makes sure that types in the bawk program are correct. For example, arrays cannot contain elements of different types; a program that attempts to insert incorrect types into an array will be halted in the semantic checking phase. Each block in the program is checked separately, but global variables in the begin block can be used in loop and end.

5.4 Code Generation

The code generator traverses the semantically-checked AST (generated by `semant`) and translates each node into LLVM code. The code generator also links the necessary C libraries and makes calls to the proper C functions when necessary.

5.5 C Libraries

Because of the complexity of some aspects of our language, we outsourced some functionality to a number of C programs and libraries which allowed us to implement higher-level processes without having to rely on complicated LLVM constructs. The figure below shows the coordination between the `structure.c` file and `codegen.ml`



`regex.c`

Since the nature of our language is to process and filter text files, one core feature we needed was regular expression matching. Due to the vast number of regex patterns, it would be an immense time-suck to implement our own regex system, and the complexity would only be compounded by the fact that wed have to write it in LLVM. Since C already has a relatively easy-to-use regex library, we wrote a small wrapper for it and directed all regex expressions through it.

POSIX regex library
The POSIX regex library is the most commonly-used regex matching library in C. In order to parse a regex expression, the POSIX library requires that the expression first be compiled into a `regex_t` data structure. This type is then passed to the `regexec` function to be executed. After some experimentation, we used the `REG_EXTENDED` (adds extended syntax) and the `REG_NOSUB` (only reports success/fail) flags to provide the most intuitive usage.

`structure.c`
In contrast to the structure of a micro-C program, bawk programs loop through files in the `LOOP` block, and execute the `BEGIN`, `END`, and `CONFIG` blocks only once. Generating this structure (and simply reading through a file) is tremendously difficult using only OCaml LLVM bindings, so we elected to write a C file that would effectively moderate the execution of the program. This file, structure.c, declared the three functions `begin()`, `loop()`, and `end()`, without actually defining them. Codegen injects the actual code into these three functions, as it is translated from the AST. `structure.c` also defines the main function, which is executed upon runtime. The main function reads in the filename from the command line, loops through the file, and calls the `loop()` function on each line. Finally, it calls the `end()` function. `structure.c` also maintains the values for `RS` and `FS`, which are declared as external global variables. Codegen reads in the values from the config block and stores them in these globals, which are then used in the actual splitting of the file, and in functions such as the line access function ($) which splits a line into its fields.

mylist.c
The bulk of array implementation is done in C using a linked list implementation we built in `mylist.c`. These linked lists are designed to be generic, so that they can hold data of any type, including other lists. This simplified our array implementation because we were able to use the same linked list implementation for all array types (int, bool, string, regex) and for nested arrays.

Each built-in function for arrays is written in C and called in the code generator. Each list keeps track of the size of the underlying type (int, string, rgx, bool), and the depth of the array (number of dimensions). The underlying type of the array is used to decide which comparator to pass into functions. The built-in array functions for C are as follows:

`initList`, which creates a list.

`addFront`, which takes in a long that represents the data and adds a node to the front of the list.

`length`, which traverses the list to return the length of the list.

`insert`, which inserts a specified element at a specified index.

`delete`, which deletes an element at a specified index.

`contains`, which checks if an element is in the array. It takes in a comparator function pointer, which is passed from the code generator based on the type of the second element.

`index_of`, which finds the first index of a specified element. It takes in a comparator function pointer, which is passed from the code generator based on the type of the second element.

6 Test Plan

6.1 Test Suite

Testing is arguably the most important part of building a compiler. Without testing your code, there is no guarantee that anything you wrote actually works. Thus, throughout development we made sure to write tests for each stage (lexer, parser, semantic checking, code generation) that pass or fail. These tests cover all the features of our language. Our tests include small tests that pinpoint specific features such as operators, loops, and built-in functions. They also include larger tests such as fibonacci and gcd to confirm that the correct output is being produced.

A listing of the tests and their respective outputs is included in the Code Listing section.

6.2 Automation

With such a large test suite, it is necessary to eventually automate the testing process. As a result, we wrote a testing script, `testall.sh`, which runs every test and reports OK if the test behaved as expected or FAILED if the test didn't behave as expected.

The testing script is included in the Code Listing section. The script can be executed within the root bawk directory by running

```
$ ./bawk.sh <program.bawk> <input.txt>
```

6.3 Examples and `.ll` output

Test 1: `tests/pass-dynamicarr2.bawk`

```
BEGIN {}

```

```
LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  delete(a, 2);
  print(int_to_string(length(a)));
}
```

Output 1: `dynamicarr2.ll`

```
; ModuleID = 'Bawk'
source_filename = "Bawk"

%Array = type { %Node*, i32, i32 }
%Node = type { i64, %Node* }

@_RS = private unnamed_addr constant [2 x i8] c"\0A\00", align 1
@RS = global i8* getelementptr inbounds ([2 x i8], [2 x i8]* @_RS, i32 0, i32 0)
@_FS = private unnamed_addr constant [2 x i8] c" \00", align 1
@FS = global i8* getelementptr inbounds ([2 x i8], [2 x i8]* @_FS, i32 0, i32 0)
@fmt = private unnamed_addr constant [4 x i8] c"%s\0A\00"

declare i32 @printf(i8*, ...)

declare i8* @int_to_string(i32)

declare i8* @bool_to_string(i1)

declare i32 @string_to_int(i8*)

declare i8* @concat(i8*, i8*)

declare i1 @comp(i8*, i8*)

declare i1 @ncomp(i8*, i8*)

declare i1 @equals(i8*, i8*)

declare i1 @nequals(i8*, i8*)

declare i8* @rgx_to_string(i8*)

declare i8* @access(i8*, i32)

declare i1 @streq(i8*, i8*)

declare i1 @strneq(i8*, i8*)

declare i1 @strless(i8*, i8*)

declare i1 @strgreater(i8*, i8*)

declare i1 @strgeq(i8*, i8*)

declare i1 @strleq(i8*, i8*)

declare i32 @numfields(i8*)

declare %Array* @initList(i64, i32)

declare %Node* @addFront(%Array*, i64)

declare i64 @getElement(%Array*, i32)

declare i32 @assignElement(%Array*, i32, i64)

declare i8 @reverseList(%Array*)

declare i32 @length(%Array*)

declare i64 @removeNode(%Array*, i32)

declare i8 @insertElement(%Array*, i32, i64)

declare i32 @compareInts(i64, i64)

declare i32 @compareBools(i64, i64)

declare i32 @compareStrs(i64, i64)

declare i1 @contains(%Array*, i64, i32 (i64, i64)*)

declare i32 @findIndexOfNode(%Array*, i64, i32 (i64, i64)*)

define void @loop(i8* %line) {
entry:
  %line1 = alloca i8*
  store i8* %line, i8** %line1
  ret void
}

define void @end() {
entry:
  %a = alloca %Array*
  %initList = call %Array* @initList(i64 ptrtoint (i1* getelementptr (i1, i1* null
      , i32 1) to i64), i32 1)
  %addFront = call %Node* @addFront(%Array* %initList, i64 1)
  %addFront1 = call %Node* @addFront(%Array* %initList, i64 0)
  %addFront2 = call %Node* @addFront(%Array* %initList, i64 0)
  %reverseList = call i8 @reverseList(%Array* %initList)
  store %Array* %initList, %Array** %a
  %a3 = load %Array*, %Array** %a
  %removeNode = call i64 @removeNode(%Array* %a3, i32 2)
  %a4 = load %Array*, %Array** %a
  %length = call i32 @length(%Array* %a4)
  %int_to_string = call i8* @int_to_string(i32 %length)
  %printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4 x
      i8]* @fmt, i32 0, i32 0), i8* %int_to_string)
  ret void
}
```

Test 2: `tests/pass-ops4.bawk`

```
BEGIN {}

LOOP {}

END {
  print(bool_to_string('[0-9]*' % '[0]*'));
  print(bool_to_string('[0-9]*' % '[0-9]*'));
  print(int_to_string(99));
  print(bool_to_string('[0-9]*' !% '[0]*'));
  print(bool_to_string('[0-9]*' !% '[0-9]*'));
  print(int_to_string(99));
  print(bool_to_string("1234" ~ '[0]'));
  print(bool_to_string("1234" ~ '[0-9]*'));
  print(int_to_string(99));
  print(bool_to_string("1234" !~ '[0]'));
  print(bool_to_string("1234" !~ '[0-9]*'));
}
```

Output 2: `ops2.ll`

```
; ModuleID = 'Bawk'
source_filename = "Bawk"

%Array = type { %Node*, i32, i32 }
%Node = type { i64, %Node* }

@_RS = private unnamed_addr constant [2 x i8] c"\0A\00", align 1
@RS = global i8* getelementptr inbounds ([2 x i8], [2 x i8]* @_RS, i32 0, i32 0)
@_FS = private unnamed_addr constant [2 x i8] c" \00", align 1
@FS = global i8* getelementptr inbounds ([2 x i8], [2 x i8]* @_FS, i32 0, i32 0)
@0 = global [7 x i8] c"'[0]*'\00"
@1 = global [9 x i8] c"'[0-9]*'\00"
@fmt = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@2 = global [9 x i8] c"'[0-9]*'\00"
@3 = global [9 x i8] c"'[0-9]*'\00"
@fmt.1 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@4 = global [7 x i8] c"'[0]*'\00"
@5 = global [9 x i8] c"'[0-9]*'\00"
@fmt.3 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@6 = global [9 x i8] c"'[0-9]*'\00"
@7 = global [9 x i8] c"'[0-9]*'\00"
@fmt.4 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmt.5 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@8 = global [6 x i8] c"'[0]'\00"
@9 = global [5 x i8] c"1234\00"
@fmt.6 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@10 = global [9 x i8] c"'[0-9]*'\00"
@11 = global [5 x i8] c"1234\00"
@fmt.7 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmt.8 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@12 = global [6 x i8] c"'[0]'\00"
@13 = global [5 x i8] c"1234\00"
```

```
@fmt.9 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@14 = global [9 x i8] c"'[0-9]*'\00"
@15 = global [5 x i8] c"1234\00"
@fmt.10 = private unnamed_addr constant [4 x i8] c"%s\0A\00"

declare i32 @printf(i8*, ...)

declare i8* @int_to_string(i32)

declare i8* @bool_to_string(i1)

declare i32 @string_to_int(i8*)

declare i8* @concat(i8*, i8*)

declare i1 @comp(i8*, i8*)

declare i1 @ncomp(i8*, i8*)

declare i1 @equals(i8*, i8*)

declare i1 @nequals(i8*, i8*)

declare i8* @rgx_to_string(i8*)

declare i8* @access(i8*, i32)

declare i1 @streq(i8*, i8*)

declare i1 @strneq(i8*, i8*)

declare i1 @strless(i8*, i8*)

declare i1 @strgreater(i8*, i8*)

declare i1 @strgeq(i8*, i8*)

declare i1 @strleq(i8*, i8*)

declare i32 @numfields(i8*)

declare %Array* @initList(i64, i32)

declare %Node* @addFront(%Array*, i64)

declare i64 @getElement(%Array*, i32)

declare i32 @assignElement(%Array*, i32, i64)

declare i8 @reverseList(%Array*)

declare i32 @length(%Array*)

declare i64 @removeNode(%Array*, i32)

declare i8 @insertElement(%Array*, i32, i64)

declare i32 @compareInts(i64, i64)

declare i32 @compareBools(i64, i64)

declare i32 @compareStrs(i64, i64)

declare i1 @contains(%Array*, i64, i32 (i64, i64)*)

declare i32 @findIndexOfNode(%Array*, i64, i32 (i64, i64)*)

define void @loop(i8* %line) {
entry:
  %line1 = alloca i8*
  store i8* %line, i8** %line1
  ret void
}

define void @end() {
entry:
  %equals = call i1 @equals(i8* getelementptr inbounds ([9 x i8], [9 x i8]* @1,
      i32 0, i32 0), i8* getelementptr inbounds ([7 x i8], [7 x i8]* @0, i32 0, i32
       0))
  %bool_to_string = call i8* @bool_to_string(i1 %equals)
  %printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4 x
       i8]* @fmt, i32 0, i32 0), i8* %bool_to_string)
  %equals1 = call i1 @equals(i8* getelementptr inbounds ([9 x i8], [9 x i8]* @3,
      i32 0, i32 0), i8* getelementptr inbounds ([9 x i8], [9 x i8]* @2, i32 0, i32
       0))
  %bool_to_string2 = call i8* @bool_to_string(i1 %equals1)
  %printf3 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
      x i8]* @fmt.1, i32 0, i32 0), i8* %bool_to_string2)
  %int_to_string = call i8* @int_to_string(i32 99)
  %printf4 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
      x i8]* @fmt.2, i32 0, i32 0), i8* %int_to_string)
  %nequals = call i1 @nequals(i8* getelementptr inbounds ([9 x i8], [9 x i8]* @5,
      i32 0, i32 0), i8* getelementptr inbounds ([7 x i8], [7 x i8]* @4, i32 0, i32
       0))
  %bool_to_string5 = call i8* @bool_to_string(i1 %nequals)
  %printf6 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
      x i8]* @fmt.3, i32 0, i32 0), i8* %bool_to_string5)
  %nequals7 = call i1 @nequals(i8* getelementptr inbounds ([9 x i8], [9 x i8]* @7,
       i32 0, i32 0), i8* getelementptr inbounds ([9 x i8], [9 x i8]* @6, i32 0,
      i32 0))
  %bool_to_string8 = call i8* @bool_to_string(i1 %nequals7)
  %printf9 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
      x i8]* @fmt.4, i32 0, i32 0), i8* %bool_to_string8)
  %int_to_string10 = call i8* @int_to_string(i32 99)
  %printf11 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
       x i8]* @fmt.5, i32 0, i32 0), i8* %int_to_string10)
  %comp = call i1 @comp(i8* getelementptr inbounds ([5 x i8], [5 x i8]* @9, i32 0,
       i32 0), i8* getelementptr inbounds ([6 x i8], [6 x i8]* @8, i32 0, i32 0))
  %bool_to_string12 = call i8* @bool_to_string(i1 %comp)
  %printf13 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
       x i8]* @fmt.6, i32 0, i32 0), i8* %bool_to_string12)
  %comp14 = call i1 @comp(i8* getelementptr inbounds ([5 x i8], [5 x i8]* @11, i32
    0, i32 0), i8* getelementptr inbounds ([9 x i8], [9 x i8]* @10, i32 0, i32
    0))
  %bool_to_string15 = call i8* @bool_to_string(i1 %comp14)
  %printf16 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
    x i8]* @fmt.7, i32 0, i32 0), i8* %bool_to_string15)
  %int_to_string17 = call i8* @int_to_string(i32 99)
  %printf18 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
    x i8]* @fmt.8, i32 0, i32 0), i8* %int_to_string17)
  %ncomp = call i1 @ncomp(i8* getelementptr inbounds ([5 x i8], [5 x i8]* @13, i32
    0, i32 0), i8* getelementptr inbounds ([6 x i8], [6 x i8]* @12, i32 0, i32
    0))
  %bool_to_string19 = call i8* @bool_to_string(i1 %ncomp)
  %printf20 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
    x i8]* @fmt.9, i32 0, i32 0), i8* %bool_to_string19)
  %ncomp21 = call i1 @ncomp(i8* getelementptr inbounds ([5 x i8], [5 x i8]* @15,
    i32 0, i32 0), i8* getelementptr inbounds ([9 x i8], [9 x i8]* @14, i32 0,
    i32 0))
  %bool_to_string22 = call i8* @bool_to_string(i1 %ncomp21)
  %printf23 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
    x i8]* @fmt.10, i32 0, i32 0), i8* %bool_to_string22)
  ret void
}
a
```

7 Lessons Learned

7.1 Ashley

I learned a lot this semester about compilers and functional programming. Above all, I learned about working in a team. I realized the importance of effectively delegating tasks and ensuring that theyre actually getting completed on time. It is equally important to efficiently manage your time with your group and keep open communication throughout the entire semester. It is also extremely important to have weekly group meetings and weekly check-ins with your TA.

I'd advise future groups to make sure to really understand what needs to get done and plan for how it will get done before jumping into coding. Jumping straight into coding is never the answer. Be sure to break down everything piece by piece so you can build up your project incrementally, and set deadlines for when you want to complete each part. Make sure to constantly test along the way as well, until you literally do not want to write another test ever again. Finally, take advantage of the TAs that will be happy to help you along the way!

7.2 Christine

This was my first time really using functional programming, so there was a steep learning curve involved in learning OCaml. I also learned a lot about working in a group, which requires dividing tasks but still knowing the entire architecture and necessitates constant and open communication. Find a team that will contribute even work (which my team was awesome at)!

I would advise future groups to think carefully about designing the parser and the overall structure of the program, since you dont want to be changing things at the last minute. Everyone says this, but start tackling codegen early because it takes a lot of time to adjust to and learn OCaml and the LLVM module (especially because the documentation is quite sparse). Outsource the hard things in C if you can.

7.3 Melanie

I learned a lot about working in a team this semester. Its important for everyone to be on the same page all the time – our group found it useful to write down the discrete tasks required of everyone each week, and to continuously check in with each other to see who needed help. One of our more challenging roadblocks was when we realized that some code needed to be changed in parser during the last fourth of the project– as a result, we had to modify a large chunk of code. Its best to make sure that your parser (and even scanner) are exactly how you want them before implementing large blocks of codegen.

It's also okay to scale back some parts of your language, but try to think about what the core functionality is before starting to implement features.

Implement the most important features first– if you dont get around to the less important ones, its ok. One of the core functions of our language is file reading and looping, (and this was probably one of the more challenging aspects of the program), so we implemented this first. Some of the features we implemented towards the end of the semester were less important, and had we not gotten them to work, it wouldnt have been a big deal.

Finally, please actually learn how pattern matching works. The number of frivolous if statements I wrote in OCaml before realizing how simple and robust pattern matching is was pretty astounding.

7.4 Victoria

This project involved a lot of group collaboration, which meant making sure that all members of the team had at least a vague idea of what everyone else was working on, even if they werent actually involved. It's important that every member of the group has a good grasp on the language and the frontend of the language, no matter who actually implements them, since this will provide a good foundation for everything that follows. It is also important for the group to know what is expected of them, which we achieved by setting concrete goals to be accomplished before every time we met.

We were very ambitious in constructing our language, which wasn't necessarily a bad thing, but it meant that we had to think critically about what was feasible and most important to our language in later parts. It would have been better to hash this out when writing our LRM instead of later in the process.

8 Code Listing

`scanner.mll`
Authors: Mel, Ashley, Christine, Victoria

```
{ open Parser }

```

```
rule token = parse
  [' ' '\t' '\r' '\n'] { token lexbuf } (* Whitespace *)
| "#"        { comment lexbuf } (* Comments *)
| "("        { LPAREN }
| ")"        { RPAREN }
| "{"        { LCURLY }
| "}"        { RCURLY }
| "["        { LSQUARE }
| "]"        { RSQUARE }
| ";"        { SEMI }
| '"'        { read_string (Buffer.create 17) lexbuf }
| "'"        { read_rgx (Buffer.create 17) lexbuf }
| "&"        { STRCAT }
| "$"        { DOLLAR }
| ","        { COMMA }
| "+"        { PLUS }
| "-"        { MINUS }
| "/"        { DIVIDE }
| "*"        { MULTIPLY }
| "="        { ASSIGN }
| "=="       { EQUALS }
| "!="       { NEQ }
| "<"        { LT }
| "<="       { LEQ }
| ">"        { GT }
| ">="       { GEQ }
| "&&"       { AND }
| "||"       { OR }
| "+="       { PLUSEQ }
| "-="       { MINUSEQ }
| "++"       { INCREMENT }
| "--"       { DECREMENT }
| "if"       { IF }
| "else"     { ELSE }
| "while"    { WHILE }
| "for"      { FOR }
| "!"        { NOT }
| "return"   { RETURN }
| "function" { FUNCTION }
| "int"      { INT }
| "bool"     { BOOL }
| "void"     { VOID }
| "string"   { STRING }
| "rgx"      { RGX }
| "%"        { RGXEQ }
| "!%"       { RGXNEQ }
| "~"        { RGXSTRCMP }
| "!~"       { RGXSTRNOT }
| "CONFIG"   { CONFIG }
| "RS"       { RS }
| "FS"       { FS }
| "NF"       { NF }
| "BEGIN"    { BEGIN }
| "LOOP"     { LOOP }
| "END"      { END }
| "true"     { TRUE }
| "false"    { FALSE }
| "in"       { IN }
| ['0'-'9']+ as lxm { LITERAL(int_of_string lxm) }
| "\'" [^'\'']* "\'" as lxm { RGX_LITERAL(lxm) }
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']* as lxm { ID(lxm) }
| eof { EOF }
| _ as char { raise (Failure("illegal character " ^ Char.escaped char)) }

and comment = parse
  '\n' { token lexbuf }
| _    { comment lexbuf }

and read_string buf =
  parse
  | '"'       { STRING_LITERAL (Buffer.contents buf) }
  | '\\' '/'  { Buffer.add_char buf '/'; read_string buf lexbuf }
  | '\\' '\\' { Buffer.add_char buf '\\'; read_string buf lexbuf }
  | '\\' 'b'  { Buffer.add_char buf '\b'; read_string buf lexbuf }
  | '\\' 'f'  { Buffer.add_char buf '\012'; read_string buf lexbuf }
  | '\\' 'n'  { Buffer.add_char buf '\n'; read_string buf lexbuf }
  | '\\' 'r'  { Buffer.add_char buf '\r'; read_string buf lexbuf }
  | '\\' 't'  { Buffer.add_char buf '\t'; read_string buf lexbuf }
  | [^ '"' '\\']+
    { Buffer.add_string buf (Lexing.lexeme lexbuf);
      read_string buf lexbuf
    }
  | _ { raise (Failure ("Illegal string character: " ^ Lexing.lexeme lexbuf)) }
  | eof { raise (Failure ("String is not terminated")) }

and read_rgx buf =
  parse
  | '''       { RGX_LITERAL (Buffer.contents buf) }
  | [^ ''']+
    { Buffer.add_string buf (Lexing.lexeme lexbuf);
      read_string buf lexbuf
    }
  | _ { raise (Failure ("Illegal string character: " ^ Lexing.lexeme lexbuf)) }
  | eof { raise (Failure ("String is not terminated")) }
```

parser.mly

Authors: Mel, Ashley, Christine

```
%{ open Ast %}

%token LPAREN RPAREN LCURLY RCURLY LSQUARE RSQUARE SEMI COMMA
%token PLUS MINUS DIVIDE MULTIPLY ASSIGN STRCAT
%token EQUALS NEQ LT LEQ GT GEQ AND OR NOT
%token PLUSEQ MINUSEQ INCREMENT DECREMENT
%token RGXEQ RGXNEQ RGXSTRCMP RGXSTRNOT
%token RETURN FUNCTION
%token CONFIG BEGIN LOOP END
%token INT BOOL VOID STRING RGX TRUE FALSE
%token RS FS NF
%token IF ELSE WHILE FOR IN
```

```
%token DOLLAR
%token <int> LITERAL
%token <string> STRING_LITERAL
%token <string> RGX_LITERAL
%token <string> ID
%token EOF

%start program
%type <Ast.program> program

%nonassoc NOELSE
%nonassoc ELSE
%right ASSIGN PLUSEQ MINUSEQ
%left OR
%left AND
%left EQUALS NEQ RGXEQ RGXNEQ RGXSTRCMP RGXSTRNOT
%left LT LEQ GT GEQ
%left PLUS MINUS STRCAT
%left MULTIPLY DIVIDE
%right NOT NEG
%right INCREMENT DECREMENT
%left LSQUARE
%right DOLLAR

%%

program:
  begin_block loop_block end_block config_block EOF { ($1, $2, $3, $4) }

begin_block:
  BEGIN LCURLY global_vars_list func_list RCURLY { (List.rev $3, List.rev $4) }

loop_block:
  LOOP LCURLY local_vars_list stmt_list RCURLY { (List.rev $3, List.rev $4) }

end_block:
  END LCURLY local_vars_list stmt_list RCURLY { (List.rev $3, List.rev $4) }

config_block:
    /* nothing */                          { [] }
  | CONFIG LCURLY config_expr_list RCURLY { (List.rev $3) }

primitive:
    STRING { String }
  | INT    { Int }
  | BOOL   { Bool }
  | RGX    { Rgx }

typ:
    VOID       { Void }
  | array_type { $1 }

array_type:
    array_type LSQUARE RSQUARE { ArrayType($1) }
  | primitive                  { $1 }

```

```
func_list:
    /* nothing */  { [] }
  | func_list func { $2 :: $1 }

global_vars_list:
    /* nothing */             { [] }
  | global_vars_list var_decl { $2 :: $1 }

local_vars_list:
    /* nothing */            { [] }
  | local_vars_list var_decl { $2 :: $1 }

func:
  FUNCTION typ ID LPAREN formals_opt RPAREN LCURLY local_vars_list stmt_list
      RCURLY
    { { fname = $3;
        formals = $5;
        ret_type = $2;
        locals = List.rev $8;
        body = List.rev $9 } }

formals_opt:
    /* nothing */ { [] }
  | formals_list  { List.rev $1 }

formals_list:
    typ ID                    { [($1, $2)] }
  | formals_list COMMA typ ID { ($3, $4) :: $1 }

var_decl:
  typ ID SEMI { ($1, $2) }

config_expr_list:
    /* nothing */                { [] }
  | config_expr_list config_expr { $2 :: $1 }

config_expr:
    RS ASSIGN expr SEMI { RSAssign($3) }
  | FS ASSIGN expr SEMI { FSAssign($3) }

stmt_list:
    /* nothing */  { [] }
  | stmt_list stmt { $2 :: $1 }

stmt:
    expr SEMI                                             { Expr $1 }
  | RETURN expr_opt SEMI                                  { Return $2 }
  | WHILE LPAREN expr RPAREN code_block                   { While($3, $5) }
  | FOR LPAREN expr SEMI expr SEMI expr RPAREN code_block { For($3, $5, $7, $9) }
  | FOR LPAREN ID IN ID RPAREN code_block                 { EnhancedFor($3, $5, $7
      ) }
  | IF LPAREN expr RPAREN code_block %prec NOELSE         { If($3, $5, Block([]))
      }
  | IF LPAREN expr RPAREN code_block ELSE code_block      { If($3, $5, $7) }

code_block: LCURLY stmt_list RCURLY { Block(List.rev $2) }
```

```

expr_opt:
    /* nothing */ { Noexpr }
  | expr          { $1 }

expr:
    LITERAL                     { Literal($1) }
  | STRING_LITERAL              { StringLiteral($1) }
  | RGX_LITERAL                 { RgxLiteral($1) }
  | TRUE                        { BoolLit(true) }
  | FALSE                       { BoolLit(false) }
  | ID                          { Id($1) }
  | expr PLUS expr              { Binop($1, Add, $3) }
  | expr MINUS expr             { Binop($1, Sub, $3) }
  | expr MULTIPLY expr          { Binop($1, Mult, $3) }
  | expr DIVIDE expr            { Binop($1, Div, $3) }
  | expr EQUALS expr            { Binop($1, Equal, $3) }
  | expr NEQ expr               { Binop($1, Neq, $3) }
  | expr LT expr                { Binop($1, Less, $3) }
  | expr LEQ expr               { Binop($1, Leq, $3) }
  | expr GT expr                { Binop($1, Greater, $3) }
  | expr GEQ expr               { Binop($1, Geq, $3) }
  | expr AND expr               { Binop($1, And, $3) }
  | expr OR expr                { Binop($1, Or, $3) }
  | expr PLUSEQ expr            { Pluseq($1, $3) }
  | expr MINUSEQ expr           { Minuseq($1, $3) }
  | expr INCREMENT              { Increment($1) }
  | expr DECREMENT              { Decrement($1) }
  | expr STRCAT expr            { Strcat($1,$3) }
  | expr RGXEQ expr             { Rgxeq($1, $3) }
  | expr RGXNEQ expr            { Rgxneq($1, $3)}
  | expr RGXSTRCMP expr         { Rgxcomp($1,$3)}
  | expr RGXSTRNOT expr         { Rgxnot($1, $3)}
  | LSQUARE actuals_opt RSQUARE  { ArrayLit($2) }
  | expr LSQUARE expr RSQUARE    { ArrayDeref($1, $3) }
  | NOT expr                    { Unop(Not, $2) }
  | LPAREN expr RPAREN          { $2 }
  | ID LPAREN actuals_opt RPAREN { Call($1, $3) }
  | NF                          { NumFields }
  | DOLLAR expr                 { Access($2) }
  | MINUS expr %prec NEG        { Unop(Neg, $2) }
  | expr ASSIGN expr            { Assign($1, $3) }

actuals_opt:
    /* nothing */ { [] }
  | actuals_list  { List.rev $1 }

actuals_list:
    expr                     { [$1] }
  | actuals_list COMMA expr { $3 :: $1 }
```

ast.ml

Authors: Victoria, Ashley

```ocaml
(* Loosely based on MicroC *)

type op = Add | Sub | Mult | Div | Equal | Neq | Less | Leq | Greater |
          Geq | And | Or

type uop = Not | Neg

type typ = Int | Bool | Void | String | Rgx | ArrayType of typ

type bind = typ * string

type expr =
    Binop of expr * op * expr
  | BoolLit of bool
  | Literal of int
  | StringLiteral of string
  | RgxLiteral of string
  | Id of string
  | Assign of expr * expr
  | Call of string * expr list
  | Unop of uop * expr
  | ArrayLit of expr list
  | ArrayDeref of expr * expr
  | Access of expr
  | Pluseq of expr * expr
  | Minuseq of expr * expr
  | Increment of expr
  | Decrement of expr
  | Strcat of expr * expr
  | Rgxeq of expr * expr
  | Rgxneq of expr * expr
  | Rgxcomp of expr * expr
  | Rgxnot of expr * expr
  | NumFields
  | Noexpr

type config_expr =
    RSAssign of expr
  | FSAssign of expr

type stmt =
    Return of expr
  | Expr of expr
  | Block of stmt list
  | While of expr * stmt
  | If of expr * stmt * stmt
  | For of expr * expr * expr * stmt
  | EnhancedFor of string * string * stmt

type func_decl = {
    ret_type : typ;
    fname    : string;
    formals  : bind list;
    locals   : bind list;
    body     : stmt list;
  }
```

```

type begin_list = bind list * func_decl list
type loop_list = bind list * stmt list
type end_list = bind list * stmt list
type config_list = config_expr list

type program = begin_list * loop_list * end_list * config_list

(* Pretty-printing functions *)

let string_of_op = function
    Add -> "+"
  | Sub -> "-"
  | Mult -> "*"
  | Div -> "/"
  | Equal -> "=="
  | Neq -> "!="
  | Less -> "<"
  | Leq -> "<="
  | Greater -> ">"
  | Geq -> ">="
  | And -> "&&"
  | Or -> "||"


let string_of_uop = function
    Neg -> "-"
  | Not -> "!"

let rec string_of_expr = function
    Binop(e1, o, e2) ->
      string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_expr e2
  | BoolLit(true) -> "true"
  | BoolLit(false) -> "false"
  | Literal(l) -> string_of_int l
  | StringLiteral(s) -> s
  | RgxLiteral(r) -> r
  | Id(s) -> s
  | Assign(v, e) -> string_of_expr v ^ " = " ^ string_of_expr e
  | Call(f, el) ->
      f ^ "(" ^ String.concat ", " (List.map string_of_expr el) ^ ")"
  (*| Rgx(r) -> r*)
  | Unop(o, e) -> string_of_uop o ^ string_of_expr e
  | ArrayLit(el) -> "[" ^ String.concat ", " (List.map string_of_expr el) ^ "]"
  | ArrayDeref(v, e) -> string_of_expr v ^ "[" ^ string_of_expr e ^ "]"
  | Access(e) -> "$" ^ string_of_expr e
  | Strcat(e, g) -> string_of_expr e ^ "&" ^ string_of_expr g
  | Rgxeq(e, e1) -> string_of_expr e ^ "%" ^ string_of_expr e1
  | Rgxneq(e, e1) -> string_of_expr e ^ "!%" ^ string_of_expr e1
  | Rgxcomp(e, e1) -> string_of_expr e ^ "~" ^ string_of_expr e1
  | Rgxnot(e, e1) -> string_of_expr e ^ "!~" ^ string_of_expr e1
  | Increment(e) -> string_of_expr e ^ "++"
  | Decrement(e) -> string_of_expr e ^ "--"
  | Pluseq(e, e1) -> string_of_expr e ^ "+=" ^ string_of_expr e1
  | Minuseq(e, e1) -> string_of_expr e ^ "-=" ^ string_of_expr e1
  | NumFields -> "NF"
  | Noexpr -> ""

let string_of_config_expr = function
    RSAssign(e) -> "RS = " ^ string_of_expr e ^ "\n"
  | FSAssign(e) -> "FS = " ^ string_of_expr e ^ "\n"

let rec string_of_stmt = function
    Return(expr) -> "return " ^ string_of_expr expr ^ ";\n";
  | Expr(expr) -> string_of_expr expr ^ ";\n";
  | Block(stmts) ->
      "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
  | While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^ string_of_stmt s
  | If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^ string_of_stmt s
  | If(e, s1, s2) ->  "if (" ^ string_of_expr e ^ ")\n" ^
      string_of_stmt s1 ^ "else\n" ^ string_of_stmt s2
  | For(e1, e2, e3, s) ->
      "for (" ^ string_of_expr e1  ^ " ; " ^ string_of_expr e2 ^ " ; " ^
      string_of_expr e3  ^ ") " ^ string_of_stmt s
  | EnhancedFor(str1, str2, s) -> "for (" ^ str1 ^ " in " ^ str2 ^ ") " ^
      string_of_stmt s

let rec string_of_typ = function
    Int -> "int"
  | Bool -> "bool"
  | Void -> "void"
  | String -> "string"
  | Rgx -> "rgx"
  | ArrayType(t) -> string_of_typ t ^ "[]"

let string_of_vdecl (t, id) = string_of_typ t ^ " " ^ id ^ ";\n"

let string_of_fdecl fdecl =
  "function " ^ string_of_typ fdecl.ret_type ^ " " ^
  fdecl.fname ^ "(" ^ String.concat ", " (List.map snd fdecl.formals) ^
  ")\n{\n" ^
  String.concat "" (List.map string_of_vdecl fdecl.locals) ^
  String.concat "" (List.map string_of_stmt fdecl.body) ^
  "}\n"

let string_of_beginBlock (globals, funcs) =
  "BEGIN {\n" ^ String.concat "" (List.map string_of_vdecl globals) ^ "\n" ^
  String.concat "\n" (List.map string_of_fdecl funcs) ^ "\n}\n"

let string_of_loopBlock (locals, stmts) =
  "LOOP {\n" ^ String.concat "" (List.map string_of_vdecl locals) ^ "\n" ^
  String.concat "\n" (List.map string_of_stmt stmts) ^ "\n}\n"

let string_of_endBlock (locals, stmts) =
  "END {\n" ^ String.concat "" (List.map string_of_vdecl locals) ^ "\n" ^
  String.concat "\n" (List.map string_of_stmt stmts) ^ "\n}\n"

let string_of_configBlock (configs) =
  "CONFIG {\n" ^ String.concat "" (List.map string_of_config_expr configs) ^ "\n}\
    n"

let string_of_program(beginBlock, loopBlock, endBlock, configBlock) =
  "" ^ string_of_beginBlock beginBlock ^ "\n" ^
  "" ^ string_of_loopBlock loopBlock ^ "\n" ^
  "" ^ string_of_endBlock endBlock ^ "\n" ^
  "" ^ string_of_configBlock configBlock
```

sast.ml

Authors: Ashley, Victoria

```
open Ast

type sexpr = typ * sx
and sx =
    SBinop of sexpr * op * sexpr
  | SBoolLit of bool
  | SLiteral of int
  | SStringLiteral of string
  | SRgxLiteral of string
  | SId of string
  | SAssign of sexpr * sexpr
  | SCall of string * sexpr list
  | SRgx of string
  | SUnop of uop * sexpr
  | SArrayLit of sexpr list
  | SArrayDeref of sexpr * sexpr
  | SAccess of sexpr
  | SPluseq of sexpr * sexpr
  | SMinuseq of sexpr * sexpr
  | SIncrement of sexpr
  | SDecrement of sexpr
  | SStrcat of sexpr * sexpr
  | SRgxeq of sexpr * sexpr
  | SRgxneq of sexpr * sexpr
  | SRgxcomp of sexpr * sexpr
  | SRgxnot of sexpr * sexpr
  | SNumFields
  | SNoexpr

type sstmt =
    SReturn of sexpr
  | SExpr of sexpr
  | SBlock of sstmt list
  | SWhile of sexpr * sstmt
  | SIf of sexpr * sstmt * sstmt
  | SFor of sexpr * sexpr * sexpr * sstmt
  | SEnhancedFor of string * string * sstmt

type sfunc_decl = {
    sret_type : typ;
    sfname    : string;
    sformals  : bind list;
    slocals   : bind list;
    sbody     : sstmt list;
  }

```

```
type sbegin_list = bind list * sfunc_decl list
type sloop_list = sfunc_decl
type send_list = sfunc_decl
type sconfig_pair = string * string (* RS * FS *)

type sprogram = sbegin_list * sloop_list * send_list * sconfig_pair

(* Pretty-printing functions *)

let rec string_of_sexpr (t, e) =
  "(" ^ string_of_typ t ^ " : " ^ (match e with
    SBinop(e1, o, e2) ->
      string_of_sexpr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_sexpr e2
  | SBoolLit(true) -> "true"
  | SBoolLit(false) -> "false"
  | SLiteral(l) -> string_of_int l
  | SStringLiteral(s) -> s
  | SRgxLiteral(r) -> r
  | SId(s) -> s
  | SAssign(v, e) -> string_of_sexpr v ^ " = " ^ string_of_sexpr e
  | SCall(f, el) ->
      f ^ "(" ^ String.concat ", " (List.map string_of_sexpr el) ^ ")"
  | SRgx(r) -> r
  | SStrcat(a, b) -> string_of_sexpr a ^ string_of_sexpr b
  | SRgxcomp(a, b) -> string_of_sexpr a ^ string_of_sexpr b
  | SRgxnot(a, b) -> string_of_sexpr a ^ string_of_sexpr b
  | SRgxeq(a, b) -> string_of_sexpr a ^ string_of_sexpr b
  | SRgxneq(a, b) -> string_of_sexpr a ^ string_of_sexpr b
  | SPluseq(a, b) -> string_of_sexpr a ^ string_of_sexpr b
  | SMinuseq(a, b) -> string_of_sexpr a ^ string_of_sexpr b
  | SIncrement(a) -> string_of_sexpr a
  | SDecrement(a) -> string_of_sexpr a
  | SUnop(o, e) -> string_of_uop o ^ string_of_sexpr e
  | SArrayLit(el) -> "[" ^ String.concat ", " (List.map string_of_sexpr el) ^ "]"
  | SArrayDeref(v, e) -> string_of_sexpr v ^ "[" ^ string_of_sexpr e ^ "]"
  | SAccess(e) -> "$" ^ string_of_sexpr e
  | SNumFields -> "NF"
  | SNoexpr -> ""
          ) ^ ")"

let string_of_config_pair (rs, fs) =
  "RS = " ^ rs ^ "\nFS = " ^ fs ^ "\n"

let rec string_of_sstmt = function
    SReturn(expr) -> "return " ^ string_of_sexpr expr ^ ";\n";
  | SExpr(expr) -> string_of_sexpr expr ^ ";\n";
  | SBlock(stmts) ->
      "{\n" ^ String.concat "" (List.map string_of_sstmt stmts) ^ "}\n"
  | SWhile(e, s) -> "while (" ^ string_of_sexpr e ^ ") " ^ string_of_sstmt s
  | SIf(e, s, SBlock([])) -> "if (" ^ string_of_sexpr e ^ ")\n" ^ string_of_sstmt
      s
  | SIf(e, s1, s2) ->  "if (" ^ string_of_sexpr e ^ ")\n" ^
      string_of_sstmt s1 ^ "else\n" ^ string_of_sstmt s2
  | SFor(e1, e2, e3, s) ->
      "for (" ^ string_of_sexpr e1  ^ " ; " ^ string_of_sexpr e2 ^ " ; " ^
      string_of_sexpr e3  ^ ") " ^ string_of_sstmt s
  | SEnhancedFor(str1, str2, s) -> "for (" ^ str1 ^ " in " ^ str2 ^ ") " ^
      string_of_sstmt s

let string_of_sfdecl fdecl =
  "function " ^ string_of_typ fdecl.sret_type ^ " " ^
  fdecl.sfname ^ "(" ^ String.concat ", " (List.map snd fdecl.sformals) ^
  ")\n{\n" ^
  String.concat "" (List.map string_of_vdecl fdecl.slocals) ^
  String.concat "" (List.map string_of_sstmt fdecl.sbody) ^
  "}\n"

let string_of_sbeginBlock (globals, funcs) =
  "BEGIN {\n" ^ String.concat "" (List.map string_of_vdecl globals) ^ "\n" ^
  String.concat "\n" (List.map string_of_sfdecl funcs) ^ "\n}\n"

let string_of_sloopBlock (loop_func) =
  "LOOP {\n" ^ String.concat "" (List.map string_of_vdecl loop_func.slocals) ^ "\n
      " ^
  String.concat "\n" (List.map string_of_sstmt loop_func.sbody) ^ "\n}\n"

let string_of_sendBlock (end_func) =
  "END {\n" ^ String.concat "" (List.map string_of_vdecl end_func.slocals) ^ "\n"
      ^
  String.concat "\n" (List.map string_of_sstmt end_func.sbody) ^ "\n}\n"

let string_of_sconfigBlock (configs) =
  "CONFIG {\n" ^ (string_of_config_pair configs) ^ "\n}\n"

let string_of_sprogram(beginBlock, loopBlock, endBlock, configBlock) =
  "" ^ string_of_sbeginBlock beginBlock ^ "\n" ^
  "" ^ string_of_sloopBlock loopBlock ^ "\n" ^
  "" ^ string_of_sendBlock endBlock ^ "\n" ^
  "" ^ string_of_sconfigBlock configBlock
```

semant.ml

Authors: Victoria

```
(* Loosely based on MicroC *)

open Ast
open Sast

module StringMap = Map.Make(String)

(* Semantic checking of the AST. Returns an SAST if successful,
   throws an exception if something is wrong.
   Check each global variable, then check each function *)

let check (begin_list, loop_list, end_list, config_list) =

  (* Verify a list of bindings has no void types or duplicate names *)
  let check_binds (kind : string) (binds : bind list) =
    List.iter (function
  (Void, b) -> raise (Failure ("illegal void " ^ kind ^ " " ^ b))
```

```
      | _ -> ()) binds;
    let rec dups = function
        [] -> ()
      | ((_,n1) :: (_,n2) :: _) when n1 = n2 ->
    raise (Failure ("duplicate " ^ kind ^ " " ^ n1))
      | _ :: t -> dups t
    in dups (List.sort (fun (_,a) (_,b) -> compare a b) binds)
  in

  (**** Check global variables ****)

  let (globals, _) = begin_list in check_binds "global" globals;


  (**** Check functions ****)
  (* Collect function declarations for built-in functions: no bodies *)
  let built_in_decls =
  let add_bind map (ftyp, name, flist) = StringMap.add name {
      ret_type = ftyp;
      fname = name;
      formals = flist;
      locals = []; body = [] } map
    in List.fold_left add_bind StringMap.empty [ (Int, "string_to_int", [(String,
       "a")]);
                               (String, "int_to_string", [(Int, "a")]);
             (String, "bool_to_string", [(Bool, "a")]);
             (String, "rgx_to_string", [(Rgx, "a")]);
                                                 (Void, "length", []);
             (Void, "print", [(String, "a")]);
             (Void, "nprint", [(String, "a")]);
                                                 (Void, "contains", []);
                                                 (Int, "index_of", []);
                                                 (Void, "insert", []);
                                                 (Void, "delete", []);
                                                 (Void, "LOOP", []);
                                                 (Void, "END", []);
                                                 (Void, "loop", []);
                                                 (Void, "end", [])]
  in



   (* Add function name to symbol table *)
  let add_func map fd =
    let built_in_err = "function " ^ fd.fname ^ " may not be defined"
    and dup_err = "duplicate function " ^ fd.fname
    and make_err er = raise (Failure er)
    and n = fd.fname (* Name of the function *)
    in match fd with (* No duplicate functions or redefinitions of built-ins *)
         _ when StringMap.mem n built_in_decls -> make_err built_in_err
       | _ when StringMap.mem n map -> make_err dup_err
       | _ ->  StringMap.add n fd map
  in
   (* Collect all function names into one symbol table *)
  let (_, functions) = begin_list in let function_decls = List.fold_left add_func
     built_in_decls functions

```

```
in

(* Return a function from our symbol table *)
let find_func s =
  try StringMap.find s function_decls
  with Not_found -> raise (Failure ("unrecognized function " ^ s))
in

let (loop_locals,loop_stmts) = loop_list in check_binds "local" loop_locals;

let (end_locals,end_stmts) = end_list in check_binds "local" end_locals;

let check_function func =
  (* Make sure no formals or locals are void or duplicates *)
  check_binds "formal" func.formals;
  check_binds "local" func.locals;

  (* Raise an exception if the given rvalue type cannot be assigned to
     the given lvalue type *)
  let check_assign lvaluet rvaluet err =
     if (lvaluet = rvaluet) then lvaluet else raise (Failure err)
  in

  (* Build local symbol table of variables for this function *)
  let symbols = List.fold_left (fun m (ty, name) -> StringMap.add name ty m) (*
      this might need to be changed later*)
                StringMap.empty (globals @ func.formals @ func.locals )
  in

  (* Return a variable from our local symbol table *)
  let type_of_identifier s =
    try StringMap.find s symbols
    with Not_found -> raise (Failure ("undeclared identifier " ^ s))
  in
  (* Return a semantically-checked expression, i.e., with a type *)
  let rec expr = function
      Literal l -> (Int, SLiteral l)
    | StringLiteral l -> (String, SStringLiteral l)
    | BoolLit l  -> (Bool, SBoolLit l)
    | RgxLiteral l -> (Rgx, SRgxLiteral l)
    | Noexpr     -> (Void, SNoexpr)
    | Id s       -> (type_of_identifier s, SId s)
    | Access(a) as acc ->
       let (a, a') = expr a in
       if a <> Int then
               raise (Failure ("incorrect access in " ^ string_of_expr acc))
       else (String, SAccess(a, a'))
    | ArrayLit(l) ->
       if List.length l > 0 then
               let typ = expr(List.nth l 0) in
               let (arraytype, _) = typ in
               let check_array e =
                       let (et, e') = expr e in (et, e')
                       in let l' = List.map check_array l in
                       let types e =
                               let (a, _)  = expr e in
```

```
                           if a <> arraytype then
                                   raise(Failure("array of different types, expected " ^
                                   string_of_typ arraytype ^ " found " ^ string_of_typ a))
                           in let _ = List.map types l in (ArrayType(arraytype), SArrayLit(l'))
       else (Void, SArrayLit([]))
    | ArrayDeref (e1, e2) as e ->
       let (arr, e1') = expr e1 and (num, e2') = expr e2 in
       if num <> Int then raise(Failure("Int expression expected in " ^ string_of_expr e))
       else let find_typ arr =
               match arr with
               ArrayType(t) -> t
             | Bool | String | Rgx | Void | Int ->
                             raise(Failure("array deref should be called on array, not " ^ string_of_typ arr))
       in (find_typ arr, SArrayDeref((arr, e1'), (num, e2')))
    | NumFields -> (Int, SNumFields)
    | Assign(NumFields, _) -> raise (Failure ("illegal assignment of NF"))
    | Assign(e1, e2) as ex ->
       let check_expr typ e =
               let (t, et') =
                       match e with
                       ArrayLit(l) ->
                               if List.length l > 0 then expr e
                               else (typ, SArrayLit([]))
                     | _ -> expr e
       in (t, et') in
       let (lt, e1') = expr e1
       in let (rt, e2') = check_expr lt e2 in
       let err = "illegal assignment " ^ string_of_typ lt ^ " = " ^ string_of_typ rt ^ " in " ^ string_of_expr ex
       in (check_assign lt rt err, SAssign((lt, e1'), (rt, e2')))
    | Unop(op, e) as ex ->
       let (t, e') = expr e in
       let ty =
               match op with
               Neg when t = Int -> Int
             | Not when t = Bool -> Bool
             | _ -> raise (Failure ("illegal unary operator " ^ string_of_uop op ^ string_of_typ t ^ " in " ^ string_of_expr ex))
       in (ty, SUnop(op, (t, e')))
    | Binop(e1, op, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       let same = t1 = t2 in
       (* Determine expression type based on operator and operand types *)
       let ty =
               match op with
               Add | Sub | Mult | Div when same && t1 = Int   -> Int
             | Equal | Neq            when same               -> Bool
             | Less | Leq | Greater | Geq when same && (t1 = Int || t1 = String) -> Bool
             | And | Or when same && t1 = Bool -> Bool
             | _ -> raise(Failure ("illegal binary operator "  ^ string_of_typ
               t1 ^ " " ^ string_of_op op ^ " " ^
                        string_of_typ t2 ^ " in " ^ string_of_expr e))
       in (ty, SBinop((t1, e1'), op, (t2, e2')))
   | Increment(a) as e ->
       let (et, e') =  expr a in
       if (et <> Int) then raise (Failure("Int expected for " ^ string_of_expr e
           ))
       else (Int, SIncrement(et, e'))
   | Decrement(a) as e ->
       let (et, e') = expr a in
       if (et <> Int) then raise (Failure("Int expected for " ^ string_of_expr e
           ))
       else (Int, SDecrement(et, e'))
   | Strcat(e1, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       if (t1 <> String || t2 <> String) then raise(Failure("Strings expected
           for " ^ string_of_expr e))
       else (String, SStrcat((t1, e1'), (t2, e2')))
   | Pluseq(e1, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       if (t1 <> Int || t2 <> Int) then raise (Failure("Ints are expected for "
           ^ string_of_expr e))
       else (Int, SPluseq((t1, e1'), (t2, e2')))
   | Minuseq(e1, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       if (t1 <> Int || t2 <> Int) then raise (Failure("Ints are expected for "
           ^ string_of_expr e))
       else (Int, SMinuseq((t1, e1'), (t2, e2')))
   | Rgxeq(e1, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       if (t1 <> Rgx || t2 != Rgx) then raise(Failure("Rgx expected for " ^
           string_of_expr e))
       else (Bool, SRgxeq((t1, e1'), (t2, e2')))
   | Rgxneq(e1, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       if (t1 <> Rgx || t2 <> Rgx) then raise(Failure("Rgx expected for " ^
           string_of_expr e))
       else (Bool, SRgxneq((t1, e1'), (t2, e2')))
   | Rgxcomp(e1, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       if ((t1 = Rgx && t2 = String) || (t1 = String && t2 = Rgx)) then
               (Bool, SRgxcomp((t1, e1'), (t2, e2')))
       else raise(Failure("Different types expected for " ^ string_of_expr e))
   | Rgxnot(e1, e2) as e ->
       let (t1, e1') = expr e1 and (t2, e2') = expr e2 in
       if ((t1 = Rgx && t2 = String) || (t1 = String && t2 = Rgx)) then
               (Bool, SRgxnot((t1, e1'), (t2, e2')))
 else raise(Failure("Different types expected for " ^ string_of_expr e))
   | Call("length", args) as length ->
        if List.length args != 1 then raise (Failure("expecting one argument for
            " ^ string_of_expr length))
        else let (et, e') = expr (List.nth args 0) in
        if (et = String || et = Bool || et = Void || et = Rgx || et = Int) then
                raise (Failure("illegal argument found " ^
                string_of_typ et ^ " arraytype expected in " ^ string_of_expr
                    length))
        else (Int, SCall("length", [(et, e')]))
  | Call ("insert", args) as insert ->
        if List.length args !=3 then raise (Failure("expecting three arguments
            for " ^ string_of_expr insert))
        else let (t1, e1') = expr (List.nth args 0) and (t2, e2') = expr(List.
            nth args 1) and (t3, e3') = expr(List.nth args 2) in
        if t2 <> Int then raise (Failure("expecting index argument for insert
            but had " ^ string_of_typ t2))
        else
                (match t1 with
                String | Bool | Void | Rgx | Int -> raise (Failure("illegal
                    argument found "  ^ string_of_typ t1 ^ " expected arraytype")
                    )
              | ArrayType(t) ->
                              if (string_of_typ t = string_of_typ(t3) && t3 !=
                                  Void)
                              then (Void, SCall("insert", [(t1, e1');(t2, e2');(
                                  t3, e3')]))
                              else raise(Failure("cannot perform insert on " ^
                                         string_of_typ t1 ^ " and " ^
                                             string_of_typ t3)))
  | Call("delete", args) as delete ->
        if List.length args != 2 then raise (Failure("expecting two arguments
            for " ^ string_of_expr delete))
        else let (t1, e1') = expr (List.nth args 0) and (t2, e2') = expr (List.
            nth args 1) in
        if t2 <> Int then raise (Failure("expecting index argument for delete
            but had " ^ string_of_typ t2))
        else (t1, SCall("delete", [(t1, e1');(t2, e2')]))
  | Call("contains", args) as contains ->
        if List.length args != 2 then raise (Failure("expecting two arguments
            for " ^ string_of_expr contains))
  else let (t1, e1') = expr (List.nth args 0) and (t2, e2') = expr (List.nth
      args 1) in
        (match t1 with
        String | Bool | Void | Rgx | Int ->
                raise (Failure("illegal argument found " ^ string_of_typ t1 ^ "
                    arraytype expected"))
      | ArrayType(t) ->
                if (string_of_typ(t) = string_of_typ(t2) && t2 != Void)
                then (Bool, SCall("contains", [(t1, e1');(t2, e2')]))
                else raise(Failure("cannot perform contains on " ^ string_of_typ
                    (t1) ^ " and " ^ string_of_typ(t2))))
  | Call("index_of", args) as index_of ->
        if List.length args != 2 then raise (Failure("expecting two arguments
            for " ^ string_of_expr index_of))
  else let (t1, e1') = expr (List.nth args 0) and (t2, e2') = expr (List.nth
      args 1) in
        (match t1 with
        String | Bool | Void | Rgx | Int ->
              raise (Failure("illegal argument found " ^ string_of_typ t1 ^ "
                  arraytype expected"))
       | ArrayType(t) ->
              if (string_of_typ(t) = string_of_typ(t2) && t2 != Void)
      then (Int, SCall("index_of", [(t1, e1');(t2, e2')]))
      else raise(Failure("cannot perform index_of on " ^ string_of_typ(t1) ^ "
          and " ^ string_of_typ(t2))))
    | Call(fname, args) as call ->
        let fd = find_func fname in
        let param_length = List.length fd.formals in
        if List.length args != param_length then
                raise (Failure ("expecting " ^ string_of_int param_length ^
                       " arguments in " ^ string_of_expr call))
        else let check_call (ft, _) e =
                let (et, e') = expr e in
                let err = "illegal argument found " ^ string_of_typ et ^
                " expected " ^ string_of_typ ft ^ " in " ^ string_of_expr e
                in (check_assign ft et err, e')
                in
                let args' = List.map2 check_call fd.formals args
                in (fd.ret_type, SCall(fname, args'))
  in

  let check_bool_expr e =
    let (t', e') = expr e
    and err = "expected Boolean expression in " ^ string_of_expr e
    in if t' != Bool then raise (Failure err) else (t', e')
  in

  (* Return a semantically-checked statement i.e. containing sexprs *)
  let rec check_stmt = function
      Expr e -> SExpr (expr e)
    | If(p, b1, b2) -> SIf(check_bool_expr p, check_stmt b1, check_stmt b2)
    | For(e1, e2, e3, st) ->
  SFor(expr e1, check_bool_expr e2, expr e3, check_stmt st)
    | EnhancedFor(s1, s2, st) ->
        let s2_type = type_of_identifier s2 in
        (match s2_type with
        Bool | Rgx | String | Int | Void ->
                raise (Failure("cannot iterate over type " ^ string_of_typ
                    s2_type))
        | ArrayType(t) ->
                if (string_of_typ t = string_of_typ (type_of_identifier s1))
                    then SEnhancedFor(s1, s2, check_stmt st)
                else raise(Failure("mismatch in " ^ string_of_typ (
                    type_of_identifier s1) ^ " and " ^ string_of_typ s2_type)))
    | While(p, s) -> SWhile(check_bool_expr p, check_stmt s)
    | Return e -> let (t, e') = expr e in
      if t = func.ret_type then SReturn (t, e')
      else raise (Failure ("return gives " ^ string_of_typ t ^ " expected " ^
                  string_of_typ func.ret_type ^ " in " ^ string_of_expr e))

    (* A block is correct if each statement is correct and nothing
      follows any Return statement.  Nested blocks are flattened. *)
    | Block sl ->
        let rec check_stmt_list = function
            [Return _ as s] -> [check_stmt s]
          | Return _ :: _   -> raise (Failure "nothing may follow a return")
          | Block sl :: ss  -> check_stmt_list (sl @ ss) (* Flatten blocks *)
          | s :: ss         -> check_stmt s :: check_stmt_list ss
          | []              -> []
        in SBlock(check_stmt_list sl)
```

```

  in (* body of check_function *)
  { sret_type = func.ret_type;
    sfname = func.fname;
    sformals = func.formals;
    slocals  = func.locals;
    sbody = match check_stmt (Block func.body) with
SBlock(sl) -> sl
    | _ -> raise (Failure ("internal error: block didn't become a block?"))
  }
in
let check_config config_list =
  (*
  let expr = function
          StringLiteral l -> (String, SStringLiteral l)
    |_ -> raise(Failure("shouldn't have this expr in config"))
  in
  *)
  let obtain_config (default_rs, default_fs) config_list =
    let obtain_config_folder (rs, fs) = function
      | RSAssign e -> (match rs with
        | Some _ -> raise (Failure "RS set twice")
        | None -> (Some (string_of_expr e), fs)
        )
      | FSAssign e -> (match fs with
        | Some _ -> raise (Failure "fS set twice")
        | None -> (fs, Some (string_of_expr e))
        )
    in
    match List.fold_left
          obtain_config_folder (None, None) config_list
    with
    | (Some rs, Some fs) -> rs, fs
    | (Some rs, None) -> rs, default_fs
    | (None, Some fs) -> default_rs, fs
    | (None, None) -> default_rs, default_fs
  in
  obtain_config ("\n", " ") config_list
in
let gen_fn name fs binds stmts =
check_function ({
  ret_type = Void;
  fname = name;
  formals = fs;
  locals = binds;
  body = stmts
})
in
((globals, List.map check_function functions),
gen_fn "loop" [(String, "line")] loop_locals loop_stmts,
gen_fn "end" [] end_locals end_stmts,
check_config config_list)
```

codegen.ml

Authors: Christine Mel

```
(* Loosely based on MicroC *)

module L = Llvm
module A = Ast
open Sast

module StringMap = Map.Make(String)

let translate (begin_block, loop_block, end_block, config_block) =

  let context = L.global_context () in

  (* Create the LLVM compilation module into which
   we will generate code *)
  let the_module = L.create_module context "Bawk" in

  (* Get types from the context *)
  let i32_t = L.i32_type    context
    and i64_t = L.i64_type    context in
        let i8_t  = L.i8_type     context
    and  i1_t       = L.i1_type     context
  and str_t = L.pointer_type ( L.i8_type context ) in
    let node_t  = let node_t = L.named_struct_type context "Node" in
                   L.struct_set_body node_t [| i64_t ; L.pointer_type node_t |]
                       false;
                   node_t in
    let node_p_t  = L.pointer_type node_t in
    let arr_t = let arr_t = L.named_struct_type context "Array" in
                    L.struct_set_body arr_t [| node_p_t ; i32_t ; i32_t |] false;
                    arr_t in
    let arr_p_t = L.pointer_type arr_t in
    let void_t  = L.void_type context in

  (* Return the LLVM type for a bawk type *)
  let ltype_of_typ = function
    A.Int   -> i32_t
  | A.Bool  -> i1_t
  | A.Void  -> void_t
  | A.String -> str_t
        | A.Rgx -> str_t
    | A.ArrayType _ -> arr_p_t
    in


  (* Array helper functions *)
    let arr_elem_type = function
        A.ArrayType t -> t
        | _ -> raise (Failure "not an array type")
    in

  let rec arr_depth = function
    A.ArrayType t -> (arr_depth t) + 1
    | _ -> 0
  in

```

```
(* Declare built-in functions *)
let printf_t : L.lltype =
  L.var_arg_function_type i32_t [| L.pointer_type i8_t |] in
let printf_func : L.llvalue =
  L.declare_function "printf" printf_t the_module in

let int_to_string_t : L.lltype =
  L.function_type str_t [| i32_t |] in
let int_to_string_func : L.llvalue =
  L.declare_function "int_to_string" int_to_string_t the_module in

let bool_to_string_t : L.lltype =
  L.function_type str_t [| i1_t |] in
let bool_to_string_func : L.llvalue =
  L.declare_function "bool_to_string" bool_to_string_t the_module in

let string_to_int_t : L.lltype =
  L.function_type i32_t [| str_t |] in
let string_to_int_func : L.llvalue =
  L.declare_function "string_to_int" string_to_int_t the_module in

let concat_t : L.lltype =
  L.function_type str_t [| str_t; str_t|] in
let concat_func : L.llvalue =
  L.declare_function "concat" concat_t the_module in

let rgxcomp_t : L.lltype =
  L.function_type i1_t [| str_t; str_t |] in
let rgxcomp_func : L.llvalue =
  L.declare_function "comp" rgxcomp_t the_module in
let rgxnot_func : L.llvalue =
  L.declare_function "ncomp" rgxcomp_t the_module in

let rgxeq_func : L.llvalue =
  L.declare_function "equals" rgxcomp_t the_module in
let rgxneq_func : L.llvalue =
  L.declare_function "nequals" rgxcomp_t the_module in

let rgx_to_str_t : L.lltype =
  L.function_type str_t [| str_t |] in
let rgx_to_str_func : L.llvalue =
  L.declare_function "rgx_to_string" rgx_to_str_t the_module in

let access_t : L.lltype =
  L.function_type str_t [| str_t; i32_t|] in
let access_func : L.llvalue =
  L.declare_function "access" access_t the_module in

let strequals_t : L.lltype =
  L.function_type i1_t [| str_t; str_t |] in
let strequals_func : L.llvalue =
  L.declare_function "streq" strequals_t the_module in

let strnequals_t : L.lltype =
  L.function_type i1_t [| str_t; str_t |] in
let strnequals_func : L.llvalue =
  L.declare_function "strneq" strnequals_t the_module in

let strless_func : L.llvalue =
  L.declare_function "strless" strequals_t the_module in

let strgreater_func : L.llvalue =
  L.declare_function "strgreater" strequals_t the_module in

let strgeq_func : L.llvalue =
  L.declare_function "strgeq" strequals_t the_module in

let strleq_func : L.llvalue =
  L.declare_function "strleq" strequals_t the_module in

let numfields_t : L.lltype =
  L.function_type i32_t [| str_t |] in
let numfields_func : L.llvalue =
  L.declare_function "numfields" numfields_t the_module in

(* array functions *)
let initlist_t : L.lltype =
  L.function_type arr_p_t [| i64_t; i32_t |] in
let initlist_func : L.llvalue =
  L.declare_function "initList" initlist_t the_module in

let addfront_t : L.lltype =
  L.function_type node_p_t [| arr_p_t; i64_t|] in
let addfront_func : L.llvalue =
  L.declare_function "addFront" addfront_t the_module in

let arrayderef_t : L.lltype =
  L.function_type i64_t [| arr_p_t; i32_t |] in
let arrayderef_func : L.llvalue =
  L.declare_function "getElement" arrayderef_t the_module in

let assign_t : L.lltype =
  L.function_type i32_t [| arr_p_t; i32_t; i64_t |] in
let assign_func : L.llvalue =
  L.declare_function "assignElement" assign_t the_module in

let reverse_t : L.lltype =
  L.function_type i8_t [| arr_p_t |] in
let reverse_func : L.llvalue =
  L.declare_function "reverseList" reverse_t the_module in

let length_t : L.lltype =
  L.function_type i32_t [| arr_p_t |] in
let length_func : L.llvalue =
  L.declare_function "length" length_t the_module in

let delete_t : L.lltype =
  L.function_type i64_t [| arr_p_t; i32_t |] in
let delete_func : L.llvalue =
  L.declare_function "removeNode" delete_t the_module in

let insert_t : L.lltype =
  L.function_type i8_t [| arr_p_t; i32_t; i64_t |] in
let insert_func : L.llvalue =
  L.declare_function "insertElement" insert_t the_module in

let compare_t : L.lltype =
  L.function_type i32_t [| i64_t ; i64_t |] in

let compare_p_t = L.pointer_type compare_t in

let compareint_func : L.llvalue =
  L.declare_function "compareInts" compare_t the_module in
let comparebools_func : L.llvalue =
  L.declare_function "compareBools" compare_t the_module in
let comparestr_func : L.llvalue =
  L.declare_function "compareStrs" compare_t the_module in

let contains_t : L.lltype =
  L.function_type i1_t [| arr_p_t ; i64_t ; compare_p_t |] in
let contains_func : L.llvalue =
  L.declare_function "contains" contains_t the_module in

let indexof_t : L.lltype =
  L.function_type i32_t [| arr_p_t ; i64_t ; compare_p_t |] in
let indexof_func : L.llvalue =
  L.declare_function "findIndexOfNode" indexof_t the_module in

(*--- Build begin block: globals ---*)
(* Create a map of global variables after creating each *)
let global_vars : L.llvalue StringMap.t =
  let global_var m (t, n) =
    let int_init = L.const_int (ltype_of_typ t) 0
    in StringMap.add n (L.define_global n int_init the_module) m
  in

  List.fold_left global_var StringMap.empty (fst begin_block)

in

let add_terminal builder instr =
  match L.block_terminator (L.insertion_block builder) with
    Some _ -> ()
    | None -> ignore (instr builder) in

(*---Build config block ---*)
let build_config_block (rs, fs) =
  let build_global_str str name =
    let const_str = L.define_global ("_" ^ name) (L.const_stringz context str)
        the_module in
    let _ = L.set_unnamed_addr true const_str;
            L.set_global_constant true const_str;
            L.set_linkage L.Linkage.Private const_str;
            L.set_alignment 1 const_str;
    in
    let get_ptr = L.const_in_bounds_gep const_str [| L.const_int i32_t 0; L.
        const_int i32_t 0 |] in
    (*let _ = raise (Failure "shid") in *)
    L.define_global name get_ptr the_module
  in
  ignore(build_global_str rs "RS");
  ignore(build_global_str fs "FS");

in

(*--- Build begin block: function declarations ---*)
let function_decls_no_loop_end : (L.llvalue * sfunc_decl) StringMap.t =
  let function_decl m fdecl =
    let formal_types =
      Array.of_list (List.map (fun (t,_) -> ltype_of_typ t) fdecl.sformals)
    and name = fdecl.sfname
    in let ftype = L.function_type (ltype_of_typ fdecl.sret_type) formal_types
        in
    StringMap.add name (L.define_function name ftype the_module, fdecl) m in

  List.fold_left function_decl StringMap.empty (snd begin_block)

in

let function_decls_loop : (L.llvalue * sfunc_decl) StringMap.t =
  let function_decl m fdecl =
    let formal_types =
      Array.of_list (List.map (fun (t,_) -> ltype_of_typ t) fdecl.sformals)
    and name = fdecl.sfname
    in let ftype = L.function_type (ltype_of_typ fdecl.sret_type) formal_types
        in
    StringMap.add name (L.define_function name ftype the_module, fdecl) m in
  function_decl function_decls_no_loop_end (loop_block) in

let function_decls : (L.llvalue * sfunc_decl) StringMap.t =
  let function_decl m fdecl =
    let formal_types =
      Array.of_list (List.map (fun (t,_) -> ltype_of_typ t) fdecl.sformals)
    and name = fdecl.sfname
    in let ftype = L.function_type (ltype_of_typ fdecl.sret_type) formal_types
        in
    StringMap.add name (L.define_function name ftype the_module, fdecl) m in
  function_decl function_decls_loop (end_block)
      in

(*--- Build function bodies defined in BEGIN block ---*)
let build_function_body fdecl =
  let (the_function, _) = StringMap.find fdecl.sfname function_decls in
  let func_builder = L.builder_at_end context (L.entry_block the_function) in
  let string_format_str builder = L.build_global_stringptr "%s\n" "fmt" builder
      in
  let nstring_format_str builder = L.build_global_stringptr "%s" "fmt" builder
      in

  let local_vars =
    let add_formal m (t, n) p =
      L.set_value_name n p; (* p = LLVM value of param from function declaration
           we created earlier, n = name from fdecl *)
      let local = L.build_alloca (ltype_of_typ t) n func_builder in
```

```
          ignore (L.build_store p local func_builder);
        StringMap.add n local m

      and add_local m (t, n) =
        let local_var = L.build_alloca (ltype_of_typ t) n func_builder
        in StringMap.add n local_var m

      in

      let formals = List.fold_left2 add_formal StringMap.empty fdecl.sformals
          (Array.to_list (L.params the_function)) in

      List.fold_left add_local formals fdecl.slocals

    in

    let lookup n = try StringMap.find n local_vars
                   with Not_found -> StringMap.find n global_vars

    in

    let rec expr builder((ty,e): sexpr) = match e with
      SStringLiteral s -> let l = L.define_global "" (L.const_stringz context s)
          the_module in
        L.const_bitcast (L.const_gep l [|L.const_int i32_t 0|]) str_t
      | SBoolLit b  -> L.const_int i1_t (if b then 1 else 0)
      | SLiteral i -> L.const_int i32_t i
      | SNoexpr -> L.const_int i32_t 0
      | SArrayLit a -> array_gen builder ty a
      | SId i -> L.build_load (lookup i) i builder
      | SAssign (e1, e2) ->
          let (_, e) = e1 in
          let rhs = expr builder e2 in
         (match e with
             SId i -> ignore (L.build_store rhs (lookup i) builder); rhs
           | SArrayDeref (ar, idx) ->
                let (ty, _) = ar in
                let arr_type =
                        match ty with
                        A.ArrayType t -> t
                      | _ -> raise(Failure "ArrayDeref assign is not an array") in
                let long =
                        match arr_type with
                        A.Int | A.Bool-> L.build_zext rhs i64_t "arrayDerefAssign"
                             builder
                      | A.String | A.Rgx -> L.build_pointercast rhs i64_t "
                          arrayDerefAssign" builder
                      | A.ArrayType _ -> L.build_pointercast rhs i64_t "
                          arrayDerefAssign" builder
                      | _ -> raise(Failure "unmatched type")
                in ignore (L.build_call assign_func [| expr builder ar; expr
                    builder idx; long |] "assignElement" builder); rhs
           | _ -> raise (Failure "No match on left"))
      | SArrayDeref (ar, idx) ->
        let (ty, _) = ar in
        let arr_type =
                match ty with
                A.ArrayType t -> t
              | _ -> raise(Failure "ArrayDeref is not an array") in
        let v = L.build_call arrayderef_func [| expr builder ar; expr builder idx
            |] "getElement" builder in
        (match arr_type with
            A.String | A.Rgx -> L.build_inttoptr v str_t "arrayDeref" builder
          | A.Int -> L.build_trunc v i32_t "arrayDeref" builder
          | A.Bool -> L.build_trunc v i1_t "arrayDeref" builder
          | A.ArrayType _ -> L.build_inttoptr v arr_p_t "arrayDeref" builder
          | _ -> raise (Failure "Cannot find array type"))
      | SBinop(e1, op, e2) ->
        let e1' = expr builder e1
        and e2' = expr builder e2 in
        let int_binop op =
          (match op with
              A.Add -> L.build_add
            | A.Sub -> L.build_sub
            | A.Mult -> L.build_mul
            | A.Div -> L.build_sdiv
            | A.And -> L.build_and
            | A.Or -> L.build_or
            | A.Equal -> L.build_icmp L.Icmp.Eq
            | A.Neq -> L.build_icmp L.Icmp.Ne
            | A.Less -> L.build_icmp L.Icmp.Slt
            | A.Leq -> L.build_icmp L.Icmp.Sle
            | A.Greater -> L.build_icmp L.Icmp.Sgt
            | A.Geq -> L.build_icmp L.Icmp.Sge
          ) e1' e2' "tmp" builder in
        let str_binop op =
          (match op with
              A.Equal -> L.build_call strequals_func [| e1'; e2' |] "streq"
                builder
            | A.Neq -> L.build_call strnequals_func [| e1'; e2' |] "strneq"
              builder
            | A.Greater -> L.build_call strgreater_func [| e1'; e2' |] "strgreater
              " builder
            | A.Less -> L.build_call strless_func [| e1'; e2' |] "strless" builder
            | A.Leq -> L.build_call strleq_func [| e1'; e2' |] "strleq" builder
            | A.Geq -> L.build_call strgeq_func [| e1'; e2' |] "strgeq" builder
            | _ -> raise(Failure "Invalid operation on string")
          ) in
        let bool_binop op =
          (match op with
              A.Equal -> L.build_icmp L.Icmp.Eq
              | A.Neq -> L.build_icmp L.Icmp.Ne
              | A.And -> L.build_and
              | A.Or -> L.build_or
              | _ -> raise (Failure "Boolean operator not supported")) e1' e2' "
                tmp" builder in
        if (L.type_of e1' = i32_t) then int_binop op
        else if (L.type_of e1' = i1_t) then bool_binop op
        else str_binop op
      | SUnop(uop, e) ->
        let e' = expr builder e in
        (match uop with
            A.Neg -> L.build_neg
          | A.Not -> L.build_not
        ) e' "tmp" builder;
      | SStrcat(e1, e2) -> L.build_call concat_func [| expr builder e1; expr
         builder e2 |] "concat" builder
      | SRgxcomp (e1, e2) -> L.build_call rgxcomp_func [| expr builder e1; expr
         builder e2 |] "comp" builder
      | SRgxnot (e1, e2) -> L.build_call rgxnot_func [| expr builder e1; expr
         builder e2 |] "ncomp" builder
      | SRgxeq (e1, e2) -> L.build_call rgxeq_func [| expr builder e1; expr
         builder e2 |] "equals" builder
      | SRgxneq (e1, e2) -> L.build_call rgxneq_func [| expr builder e1; expr
         builder e2 |] "nequals" builder
      | SCall ("rgx_to_string", [e]) -> L.build_call rgx_to_str_func [| expr
         builder e |] "rgx_to_string" builder
      | SCall ("int_to_string", [e]) -> L.build_call int_to_string_func [| expr
         builder e |] "int_to_string" builder
      | SCall ("string_to_int", [e]) -> L.build_call string_to_int_func [| expr
         builder e |] "string_to_int" builder
      | SCall ("bool_to_string", [e]) -> L.build_call bool_to_string_func [| expr
         builder e |] "bool_to_string" builder
      | SCall ("print", [e]) ->
        L.build_call printf_func [| string_format_str builder; (expr builder e) |]
            "printf" builder
      | SCall ("nprint", [e]) ->
        L.build_call printf_func [| nstring_format_str builder; (expr builder e)
           |] "printf" builder
      | SCall ("length", [e]) ->
        L.build_call length_func [| expr builder e |] "length" builder
      | SCall ("delete", [e1; e2]) ->
        L.build_call delete_func [| expr builder e1 ; expr builder e2 |] "
           removeNode" builder
      | SCall ("insert", [e1; e2; e3]) ->
        L.build_call insert_func [| expr builder e1 ; expr builder e2 ;
           cast_unsigned builder e3|] "insertElement" builder
      | SCall ("contains", [e1; e2]) ->
        L.build_call contains_func [| expr builder e1 ; cast_unsigned builder e2 ;
            choose_compar builder e2 |] "contains" builder
      | SCall ("index_of", [e1 ; e2]) ->
          L.build_call indexof_func [| expr builder e1 ; cast_unsigned builder e2
            ; choose_compar builder e2 |] "findIndexOfNode" builder
      | SCall (f, args) ->
        let (fdef, fdecl) = StringMap.find f function_decls in
        let llargs = List.rev (List.map (expr builder) (List.rev args)) in
        let result = (match fdecl.sret_type with
                        A.Void -> ""
                      | _ -> f ^ "_result") in
         L.build_call fdef (Array.of_list llargs) result builder
      | SAccess (a) ->
         let (loop_func, _) = StringMap.find "loop" function_decls in
         L.build_call access_func [| L.param loop_func 0; expr builder a|] "access
            " builder

      | SNumFields ->
          let (loop_func, _) = StringMap.find "loop" function_decls in
          L.build_call numfields_func [| L.param loop_func 0 |] "numfields"
```

```
              builder
      | SIncrement(e) -> let e2 = (A.Int, SAssign(e, (A.Int, SBinop(e, A.Add, (A.
          Int, SLiteral(1)))))) in expr builder e2
      | SDecrement(e) -> let e2 = (A.Int, SAssign(e, (A.Int, SBinop(e, A.Sub, (A.
          Int, SLiteral(1)))))) in expr builder e2
      | SPluseq(e1, e2) -> let e = (A.Int, SAssign(e1, (A.Int, SBinop(e1, A.Add,
          e2)))) in expr builder e
      | SMinuseq(e1, e2) -> let e = (A.Int, SAssign(e1, (A.Int, SBinop(e1, A.Sub,
          e2)))) in expr builder e
      | SRgxLiteral s -> let l = L.define_global "" (L.const_stringz context s)
          the_module in
         L.const_bitcast (L.const_gep l [|L.const_int i32_t 0|]) str_t
      | _ -> raise (Failure "Cannot find expression")
               and

    (* array gen functions *)
    find_arr_type ty =
      let ast_typ = arr_elem_type ty in
      L.size_of (ltype_of_typ ast_typ)

    and get_depth ty =
      let depth_int = arr_depth ty in L.const_int i32_t depth_int

    and array_gen builder ty arr =
      let arr_type = find_arr_type ty in (* type of 1 depth down *)
      let depth = get_depth ty in

      let lst =
        L.build_call initlist_func [| arr_type; depth |] "initList" builder
      in

      let add_front e =
        let red_expr = expr builder e in
        let ty = L.type_of red_expr in
        let data = match  L.classify_type ty with
            L.TypeKind.Pointer -> L.build_pointercast red_expr i64_t "addFrontCast
                " builder
          | L.TypeKind.Integer -> L.build_zext red_expr i64_t "addFrontCast"
              builder
          | _ -> raise (Failure "Cannot find type of array")
        in

        ignore(L.build_call addfront_func [| lst; data |] "addFront" builder)
    in
    List.iter add_front arr; ignore(L.build_call reverse_func [| lst |] "
        reverseList" builder); lst

    and cast_unsigned builder e3 =
      let red_expr = expr builder e3 in
      let ty = L.type_of red_expr in match (L.classify_type ty) with
          L.TypeKind.Pointer -> L.build_pointercast red_expr i64_t "castToUnsigned
              " builder
        | L.TypeKind.Integer -> L.build_sext_or_bitcast red_expr i64_t "
            castToUnsigned" builder
        | _ -> raise (Failure "Cannot cast data to i64")

    and choose_compar builder e2 =
      let (e2_ty, _) = e2 in
      let rec compar_from_typ ty = match ty with
          A.Int -> L.build_zext_or_bitcast compareint_func compare_p_t "
              compareCast" builder
        | A.Bool -> L.build_zext_or_bitcast comparebools_func compare_p_t "
            compareCast" builder
        | A.String -> L.build_zext_or_bitcast comparestr_func compare_p_t "
            compareCast" builder
        | A.Rgx -> L.build_zext_or_bitcast comparestr_func compare_p_t "
            compareCast" builder
        | A.ArrayType t -> compar_from_typ t
        | _ -> raise (Failure "Unable to find comparator for list")
      in compar_from_typ e2_ty

  in


    let rec stmt builder = function
        SExpr ex -> ignore(expr builder ex); builder
      | SBlock sl -> List.fold_left stmt builder sl
      | SReturn e -> ignore (match fdecl.sret_type with
            A.Void -> L.build_ret_void builder
          | _ -> L.build_ret (expr builder e) builder); builder
      | SIf (predicate, then_stmt, else_stmt) ->
        let bool_val = expr builder predicate in
        let merge_bb = L.append_block context "merge" the_function in
            let build_br_merge = L.build_br merge_bb in (* partial function *)

        let then_bb = L.append_block context "then" the_function in
        add_terminal (stmt (L.builder_at_end context then_bb) then_stmt)
        build_br_merge;

        let else_bb = L.append_block context "else" the_function in
        add_terminal (stmt (L.builder_at_end context else_bb) else_stmt)
        build_br_merge;

        ignore(L.build_cond_br bool_val then_bb else_bb builder);
        L.builder_at_end context merge_bb
      | SWhile (predicate, body) ->
        let pred_bb = L.append_block context "while" the_function in
        ignore(L.build_br pred_bb builder);

        let body_bb = L.append_block context "while_body" the_function in
        add_terminal (stmt (L.builder_at_end context body_bb) body)
              (L.build_br pred_bb);

        let pred_builder = L.builder_at_end context pred_bb in
        let bool_val = expr pred_builder predicate in

        let merge_bb = L.append_block context "merge" the_function in
        ignore(L.build_cond_br bool_val body_bb merge_bb pred_builder);
        L.builder_at_end context merge_bb
      | SFor (e1, e2, e3, body) -> stmt builder
        ( SBlock [SExpr e1 ; SWhile (e2, SBlock [body ; SExpr e3]) ] )
      | _ -> raise (Failure "Cannot pattern match statement")
    in

    let func_builder = stmt func_builder (SBlock fdecl.sbody) in

    (* return types for functions *)
    add_terminal func_builder (match fdecl.sret_type with
        A.Void -> L.build_ret_void
      | A.Int -> L.build_ret (L.const_int i32_t 0)
      | A.Bool -> L.build_ret (L.const_int i1_t 0)
      | A.String -> L.build_ret (L.const_pointer_null str_t)
      | t -> L.build_ret (L.const_int (ltype_of_typ t) 0))

  (*----- END OF build_func_body ----- *)

  in

  (* Call the things that happen in main *)
  build_config_block config_block;
  List.iter build_function_body (snd begin_block);
  build_function_body loop_block;
  build_function_body end_block;

  the_module
```

structure.c

Author: Mel

```
/* Allows LOOP block to loop through a file */

#include <stdio.h>
#include <string.h>

extern char *RS;
extern char *FS;

void setRS(char *newRS) {
  RS = newRS;
}

void setFS(char *newFS) {
  FS = newFS;
}

void loop(char *line);

void end();

int main(int argc, char **argv) {

  /* Error handling for if filename is not specified */
  if (argc < 2) {
    fprintf(stderr, "usage: ./bawk.sh [bawk file] [input file]\n");
  }
  else {
    char *filename = argv[1];
    FILE *fp = fopen(filename, "rw");
    char buffer[256];
    size_t n = sizeof(buffer);
    char *buf = buffer;

    while(getdelim(&buf, &n, *RS, fp) > 0){
      buffer[strcspn(buffer, RS)] = '\0';
      loop(buffer);
    }

    end();
  }
}
```

mylist.c

Author: Ashley

```
/*
 * Arrays
 * Loosely based on Lab 3 from COMS 3157
 */

#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>

/* A node in a linked list */
struct Node {
  unsigned long data;
  struct Node *next;
};

/* A linked list. 'head' points to the first node in the list */
struct List {
  struct Node *head;
  size_t size_of_type;
  int depth;
};

/* Traverse list to find length of list */
int length(struct List *list) {
  struct Node *node = list->head;
  int count = 0;
  while(node) {
    node = node->next;
    count++;
  }
  return count;
}

/* Compare two bools. Return -1 if a < b, return 1 if a > b, and return 0 if a ==
    b */
```

```
int compareBools(unsigned long a, unsigned long b)
{
  if ((bool)a < (bool)b) return -1;
  if ((bool)a > (bool)b) return 1;
  return 0;
}

/* Compare two ints. Return -1 if a < b, return 1 if a > b, and return 0 if a == b
    */
int compareInts(unsigned long a, unsigned long b)
{
  if ((int)a < (int)b) return -1;
  if ((int)a > (int)b) return 1;
  return 0;
}

/* Compare two strings. Return -1 if a < b, return 1 if a > b, and return 0 if a
    == b */
int compareStrs(unsigned long a, unsigned long b)
{
  return strcmp((const char *)a, (const char *)b);
}

/* Compare two lists. Return -1 if a < b, return 1 if a > b, and return 0 if a ==
    b */
int compareLists(unsigned long a, unsigned long b, int (*compar)(unsigned long,
    unsigned long))
{
  struct List *lista = (struct List *)a;
  struct List *listb = (struct List *)b;
  if ( length(lista) != length(listb) )
    return -1;
  if ( lista->depth > 1 ) {
    struct Node *nodea = lista->head;
    struct Node *nodeb = listb->head;
    int total = 0;
    while( nodea ) {
      total += abs(compareLists(nodea->data, nodeb->data, compar));
      nodea = nodea->next;
      nodeb = nodeb->next;
    }
    if ( total == 0 )
      return 0;
    else
      return -1;
  }
  if ( lista->depth == 1) {
    struct Node *nodea = lista->head;
    struct Node *nodeb = listb->head;
    int total = 0;
    while( nodea ) {
      total += abs(compar(nodea->data, nodeb->data));
      nodea = nodea->next;
      nodeb = nodeb->next;
    }
    if ( total == 0 )
```

```
      return 0;
    else
      return -1;
  }
  return -1;
}

/* Initialize and return an empty list */
struct List *initList(size_t size_of_type, int depth)
{
  struct List *list = malloc( sizeof(struct List) );
  if(list == NULL){
    perror("malloc returned NULL");
    exit(1);
  }
  list->head = 0;
  list->size_of_type = size_of_type;
  list->depth = depth;
  return list;
}

/* Traverse the list until node at index found. NULL if not found */
struct Node *findByIndex(struct List *list, int indexSought)
{
  int arr_len = length(list);

  struct Node *node = list->head;
  int indexAt = 0;
  while(node) {
    if( indexSought == indexAt )
      return node;
    node = node->next;
    indexAt++;
  }
  return NULL;
}

/*
 * Traverse the list, comparing each data item with 'dataSought' using
 * 'compar' function.  ('compar' returns 0 if the data pointed to by
 * the two parameters are equal, non-zero value otherwise.)
 *
 * Returns the first node containing the matching data,
 * NULL if not found.
 */
struct Node *findNode(struct List *list, unsigned long dataSought, int (*compar)(
    unsigned long, unsigned long))
{
  struct Node *node = list->head;
  if(list->depth == 1) {
    while(node) {
      if( compar(dataSought, node->data) == 0 )
        return node;
      node = node->next;
    }
  }
  if(list->depth > 1) {
    while(node) {
      if( compareLists(dataSought, node->data, compar) == 0 )
        return node;
      node = node->next;
    }
  }
  return NULL;
}

/* Returns 1 if dataSought is found in list, and 0 otherwise */
int contains(struct List *list, unsigned long dataSought, int (*compar)(unsigned
    long, unsigned long)) {
  struct Node *found = findNode(list, dataSought, compar);
  if (found)
    return 1;
  return 0;
}

/* Traverse list to find index of node */
int findIndexOfNode(struct List *list, unsigned long dataSought, int (*compar)(
    unsigned long, unsigned long))
{
  struct Node *node = list->head;
  int count = 0;
  if(list->depth == 1) {
    while(node) {
      if( compar(dataSought, node->data) == 0 ) {
        return count;
      }
      node = node->next;
      count++;
    }
  }
  if(list->depth > 1) {
    while(node) {
      if( compareLists(dataSought, node->data, compar) == 0 ) {
        return count;
      }
      node = node->next;
      count++;
    }
  }
  return count;
}

/*
 * Remove node at specific index and return the 'data' pointer that was stored in
     the node.
 * Returns NULL if the list is empty
 */
unsigned long removeNode(struct List *list, int indexSought)
{
  struct Node *node = list->head;
  int indexAt = 0;

  int arr_len = length(list);
  if (indexSought >= arr_len || indexSought < 0) {
    fprintf(stderr, "%s\n", "Index Out of Bounds Error.");
    exit(0);
  }

  // check that list is not empty
  if ( list->head ) {
    // if remove first index, it's just like popFront()
    if ( indexSought == 0 ) {
      unsigned long data = list->head->data;
      struct Node *node = list->head;
      list->head = list->head->next;
      free(node);
      return data;
    }

    // remove index other than first index
    while(node) {
      if( indexSought - 1 == indexAt ) {
        unsigned long data = node->next->data;
        struct Node *deletedNode = node->next;
        node->next = node->next->next;
        free(deletedNode);
        return data;
      }
      node = node->next;
      indexAt++;
    }
  }
  return 0;
}

/*
 * Remove the first node from the list, and return the 'data' pointer that was
     stored in the node.
 * Returns NULL is the list is empty.
 */
unsigned long popFront(struct List *list)
{
  if(list->head) {
    unsigned long data = list->head->data;
    struct Node *node = list->head;
    list->head = list->head->next;
    free(node);
    return data;
  }
  return 0;
}

/* Remove all nodes from the list */
void removeAllNodes(struct List *list)
{
  while(list->head) {
    popFront(list);
  }
```

```c
}

/*
 * Create a node that holds the given data pointer,
 * and add the node to the front of the list.
 * Returns the newly created node on success and NULL on failure.
 */
struct Node *addFront(struct List *list, unsigned long data)
{
  struct Node *node = malloc( sizeof(struct Node) );
  if(node == NULL){
    perror("malloc returned NULL");
    exit(1);
  }
  node->next = list->head;
  node->data = data;
  list->head = node;
  return node;
}

/*
 * Create a node that holds the given data pointer,
 * and add the node right after the node passed in as the 'prevNode'
 * parameter.  If 'prevNode' is NULL, this function is equivalent to
 * addFront().
 * Returns the newly created node on success and NULL on failure.
 */
struct Node *addAfter(struct List *list, struct Node *prevNode, unsigned long data
    )
{
  struct Node *node = malloc( sizeof(struct Node) );
  if( prevNode ) {
    node->next = prevNode->next;
    node->data = data;
    prevNode->next = node;
  }
  else {
    node->next = list->head;
    node->data = data;
    list->head = node;
  }
  return node;
}

/* Reverse the list. */
void reverseList(struct List *list)
{
  struct Node *prv = NULL;
  struct Node *cur = list->head;
  struct Node *nxt;

  while (cur) {
    nxt = cur->next;
    cur->next = prv;
    prv = cur;
    cur = nxt;
  }

  list->head = prv;
}

/* Get element at specified index from a list */
unsigned long getElement(struct List *list, int index)
{
  struct Node *node_by_index = findByIndex(list, index);
  int arr_len = length(list);
  if (index >= arr_len || index < 0) {
    fprintf(stderr, "%s\n", "Index Out of Bounds Error.");
    exit(0);
  }
  return node_by_index->data;
}

/* Insert element in list at specified index */
void insertElement(struct List *list, int index, unsigned long insert)
{
  int arr_len = length(list);
  if (index > arr_len || index < 0) {
    fprintf(stderr, "%s\n", "Index Out of Bounds Error.");
    exit(0);
  }
  if (index == 0)
    addFront(list, insert);
  else {
    int x = index - 1;
    struct Node *node = findByIndex(list, x);
    addAfter(list, node, insert);
  }
}

/* Assign element at specified index to a new value */
void assignElement(struct List *list, int index, unsigned long insert)
{
  struct Node *node = findByIndex(list, index);
  node->data = insert;
}
```

rgx.c
Author: Mel

```
/* Regex operators */

#include <regex.h>
#include <stdlib.h>
#include <stdio.h>
#include <assert.h>
#include <string.h>

/* Returns 1 if string a matches rgx b, 0 otherwise */
int comp(char *a, char *b) {
  if (b[0] == '\'') {
    b++;
  }
  if (b[strlen(b) - 1] == '\'') {
    b[strlen(b) - 1] = 0;
  }
  regex_t regex;
  int comp = regcomp(&regex, b, REG_EXTENDED|REG_NOSUB);
  if (comp) {
    printf("Regex expression could not be compiled");
    return 0;
  }
  comp = regexec(&regex, a, 0, NULL, 0);
  if (!comp) {
    return 1;
  }
  return 0;
}

/* Returns 1 if string a doesn't match rgx b, 0 otherwise */
int ncomp(char *a, char *b) {
  int opp = comp(a, b);
  return !opp;
}

/* rgx a == rgx b */
int equals(char *a, char *b) {
  if (strcmp(a, b) == 0) {
    return 1;
  }
  return 0;
}

/* rgx a != rgx b */
int nequals(char *a, char *b) {
  return !equals(a, b);
}
```

convert.c

Author: Mel

```
/* Built-in conversion functions, string operations, NF, $ */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

extern char *FS;

/* Convert a string to an int */
int string_to_int(char *a) {
  char *cleaned = malloc(strlen(a) + 1);
  strcpy(cleaned, a);
  return atoi(cleaned);
```

```
}

/* string a == string b */
int streq(char *a, char *b) {
  return (strcmp(a, b) == 0);
}

/* string a != string b */
int strneq(char *a, char *b) {
  return (strcmp(a, b) != 0);
}

/* string a > string b */
int strgreater(char *a, char *b) {
  return (strcmp(a, b) > 0);
}

/* string a < string b */
int strless(char *a, char *b) {
  return (strcmp(a, b) < 0);
}

/* string a >= string b */
int strgeq(char *a, char *b) {
  return (strcmp(a, b) >= 0);
}

/* string a <= string b */
int strleq(char *a, char *b) {
  return (strcmp(a, b) <= 0);
}

/* Concatenate string b to string a and then return the new string */
char *concat(char *a, char *b) {
  char *str = malloc(strlen(a) + strlen(b));
  size_t len1 = strlen(a), len2 = strlen(b);
  char *concat = (char*) malloc(len1 + len2 + 1);

  memcpy(concat, a, len1);
  memcpy(concat+len1, b, len2+1);
  return concat;
}

/* Convert a bool to a string */
char *bool_to_string(int a) {
  if(a) {
    return "true";
  }
  else {
    return "false";
  }
  return "";
}

/* Convert an int to a string */
char *int_to_string(int a) {
  int length = snprintf( NULL, 0, "%d", a);
  char* str = malloc( length + 1 );
  snprintf( str, length + 1, "%d", a );
  return str;
}

/* Convert a regex to a string */
char *rgx_to_string(char *a) {
  if (a[0] == '\'') {
    a++;
  }
  if (a[strlen(a)-1] == '\'') {
    a[strlen(a)-1] = '\0';
  }
  return a;
}

/* Return the number of fields in a line */
int numfields(char *line) {
    if (strlen(line)==1 && line[0] == '\n') {
      return 0;
    }
    if (strlen(line)==0) {
      return 0;
    }
    int numSpaces = 1;

    for(int i = 0; i < strlen(line); i++) {
      if(line[i] ==' '){
        numSpaces++;
      }
    }
    return numSpaces;
}

/* Return whatever is at the specified field */
char *access(char *line, int field) {
   if (field == 0) {
     return line;
   }
   char *copy = malloc(strlen(line) + 1);
   strcpy(copy, line);
   char *token;
   int count = 1;

   token = strtok(copy, FS);

   /* walk through other tokens */
   while( token != NULL ) {
      if (count == field) {
        return token;
      }
      token = strtok(NULL, FS);
      count++;
   }
   return "";
```

```
}
```

testall.sh

Author: Ashley

```
LLI="lli"
LLC="llc"
CC="cc"
BAWK="./bawk.native"

ulimit -t 30

globallog=testall.log
rm -f $globallog
error=0
globalerror=0

keep=0

Usage() {
  echo "Usage: testall.sh [options] [.bawk files]"
  echo "-k    Keep intermediate files"
  echo "-h    Print this help"
  exit 1
}

SignalError() {
    if [ $error -eq 0 ] ; then
  echo "FAILED"
  error=1
    fi
    echo "  $1"
}

Compare() {
    generatedfiles="$generatedfiles $3"
    echo diff -b $1 $2 ">" $3 1>&2
    diff -b "$1" "$2" > "$3" 2>&1 || {
  SignalError "$1 differs"
  echo "FAILED $1 differs from $2" 1>&2
    }
}

Run() {
    echo $* 1>&2
    eval $* || {
  SignalError "$1 failed on $*"
  return 1
    }
}

RunFail() {
    echo $* 1>&2
    eval $* && {
```

```
  SignalError "failed: $* did not report an error"
  return 1
    }
    return 0
}

Check() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                             s/.bawk//'`
    reffile=`echo $1 | sed 's/.bawk$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe $
        {basename}.out" &&
    Run "$BAWK" "-c" "$1" ">" "${basename}.ll" &&
    Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s" &&
    Run "$CC" "-o" "${basename}.exe" "${basename}.s" "convert.o" "structure.o" "
        mylist.o" "rgx.o" &&
    Run "./${basename}.exe" "input.txt" "&>" "${basename}.out" &&
    Compare ${basename}.out ${reffile}.out ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
  if [ $keep -eq 0 ] ; then
      rm -f $generatedfiles
  fi
  echo "OK"
  echo "###### SUCCESS" 1>&2
    else
  echo "###### FAILED" 1>&2
  globalerror=$error
    fi
}

CheckFail() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                             s/.bawk//'`
    reffile=`echo $1 | sed 's/.bawk$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    RunFail "$BAWK" "-c" "<" $1 "2>" "${basename}.err" ">>" $globallog &&
    Compare ${basename}.err ${reffile}.err ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
  if [ $keep -eq 0 ] ; then
      rm -f $generatedfiles
  fi
  echo "OK"
  echo "###### SUCCESS" 1>&2
    else
  echo "###### FAILED" 1>&2
  globalerror=$error
    fi
}

while getopts kdpsh c; do
    case $c in
  k) # Keep intermediate files
      keep=1
      ;;
  h) # Help
      Usage
      ;;
    esac
done

shift `expr $OPTIND - 1`

LLIFail() {
  echo "Could not find the LLVM interpreter \"$LLI\"."
  echo "Check your LLVM installation and/or modify the LLI variable in testall.sh"
  exit 1
}

which "$LLI" >> $globallog || LLIFail

if [ $# -ge 1 ]
then
    files=$@
else
    files="tests/pass-*.bawk tests/fail-*.bawk"
fi

for file in $files
do
    case $file in
  *pass-*)
      Check $file 2>> $globallog
      ;;
  *fail-*)
      CheckFail $file 2>> $globallog
      ;;
  *)
      echo "unknown file type $file"
      globalerror=1
      ;;
    esac
done

exit $globalerror
```

`bawk.sh`
Author: Mel

```
set -e
./bawk.native -c $1 > bawk_out.ll
llc -relocation-model=pic bawk_out.ll > bawk_out.s
cc -o bawk_out.exe bawk_out.s convert.o structure.o mylist.o rgx.o
./bawk_out.exe $2
```

All tests
Author: Ashley
`fail-FS.bawk`

```
BEGIN {}

LOOP {}

END {
  FS = "  "; # Fail: reassigning FS
}

CONFIG {
  FS = " ";
}
```

`fail-RS.bawk`

```
BEGIN {}

LOOP {}

END {
  RS = "\n\n"; # Fail: reassigning RS
}

CONFIG {
  RS = "\n";
}
```

`fail-array1.bawk`

```
BEGIN {}

LOOP {}

END {
  void[] a; # Fail: Creating an array of voids
  a = [];
}
```

fail-array2.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  bool b;
  a = [1,2,3];
  b = contains(a, true); # Fail: Checking for bool inside int array
}
```

fail-array3.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  int b;
  a = [1,2,3];
  b = index_of(a, true); # Fail: Finding index of bool in an int array
}
```

fail-arrayassign1.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [[1,2], [3,4]]; # Fail: assigning a 2D array to a 1D array
}
```

fail-arrayassign2.bawk

```
BEGIN {}

```

```
LOOP {}

END {
  int[] a;
  a = [1, "hi"]; # Fail: assigning a string as an element in an int array
}
```

fail-arrayassign3.bawk

```
BEGIN {}

LOOP {}

END {
  int[][] a;
  a = [[1,2], ["hi", "bye"]]; # Fail: Assigning 2D array of int type and string
      type to 2D array of int type
}
```

fail-assign1.bawk

```
BEGIN {}

LOOP {}

END {
  int i;
  bool b;

  i = 42;
  i = 10;
  b = true;
  b = false;
  i = false; # Fail: assigning a bool to an integer
}
```

fail-assign2.bawk

```
BEGIN {}

LOOP {}

END {
  int i;
  bool b;

  b = 48; # Fail: assigning an integer to a bool
}
```

fail-assign3.bawk

```
BEGIN {
  function void myvoid()
  {
    return;
  }
}

LOOP {}

END {
  int i;

  i = myvoid(); # Fail: assigning a void to an integer
}
```

fail-assign4.bawk

```
BEGIN {}

LOOP {}

END {
  void x; # Fail: Declaring a variable with void type
}
```

fail-assign5.bawk

```
BEGIN {}

LOOP {}

END {
  string s;
  s = '[0]*'; # Fail: assigning regex to string
}
```

fail-assign6.bawk

```
BEGIN {}

LOOP {}

END {
  NF = 3; # Fail: assigning a value to NF
}
```

fail-assign7.bawk

```
BEGIN {}

```

```
LOOP {}

END {
  rgx r;
  r = "hello"; # Fail: assigning string to regex
}
```

fail-blocks1.bawk

```
BEGIN {}

BEGIN {} # Fail: two BEGIN blocks

LOOP {}

END {}
```

fail-conversion1.bawk

```
BEGIN {}

LOOP {
  print(int_to_string(string_to_int(8))); # Fail: Passing an int instead of string
       to string_to_int
}

END {}
```

fail-conversion2.bawk

```
BEGIN {}

LOOP {}

END {
  print(int_to_string("hi")); # Fail: Passing a string instead of int
}
```

fail-conversion3.bawk

```
BEGIN {}

LOOP {}

END {
  print(rgx_to_string(2)); # Fail: Passing a int instead of rgx
}
```

fail-conversion4.bawk

```
BEGIN {}

LOOP {}

END {
  print(bool_to_string('[0-9]*')); # Fail: Passing a rgx instead of bool
}
```

fail-conversion5.bawk

```
BEGIN {}

LOOP {}

END {
  print(bool_to_string('[0-9]*')); # Fail: Passing a rgx instead of bool
}
```

fail-dead1.bawk

```
BEGIN {
  function int foo(){
    int i;

    i = 15;
    return i;
    i = 32; # Error: code after a return
  }
}

LOOP {}

END {}
```

fail-decl1.bawk

```
BEGIN {}

LOOP {}

END {
  int NF; # Fail: declaring a variable called NF
}
```

fail-decl2.bawk

```
BEGIN {}

LOOP {}

```

```
END {
  int RS; # Fail: declaring RS
}
```

fail-decl3.bawk

```
BEGIN {}

LOOP {}

END {
  int FS; # Fail: declaring FS
}
```

fail-dollarbegin.bawk

```
BEGIN {
  string hello;
  hello = $0;
}

LOOP {}

END {}
```

fail-dynamicarr1.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1,2,3];
  insert(a, 1, true); # Fail: Inserting bool into int array
}
```

fail-expr1.bawk

```
BEGIN {}

LOOP {}

END {
  int i;
  bool b;

  i = 42;
  i = 10;
  b = true;
  b = false;
  i = false; # Fail: assigning a bool to an integer
}
```

fail-expr2.bawk

```
BEGIN {
  int a;
  bool b;

  function void foo(int c, bool d)
  {
    int dd;
    bool e;
    a + c;
    c - a;
    a * 3;
    c / 2;
    d + a; # Error: bool + int
  }
}

LOOP {}

END {}
```

fail-expr3.bawk

```
BEGIN {
  int a;
  bool b;

  function void foo(int c, bool d)
  {
    int d;
    bool e;
    b + a; # Error: bool + int
  }
}

LOOP {}

END {}
```

fail-expr4.bawk

```
BEGIN {
  int a;
  string b;

  function void foo(int c, string d)
  {
    int d;
    string e;
    b + a; # Error: string + int
  }
}

LOOP {}

END {}
```

fail-expr5.bawk

```
BEGIN {}

LOOP {}

END {
  '[0]*' ~ true; # Fail: Comparing rgx and bool
}
```

fail-for1.bawk

```
BEGIN {}

LOOP {}

END {
  int i;

  for (i = 0 ; i < 10 ; i = i + 1) {
    if (i == 3) {
      print(int_to_string(42));
    }
  }

  for (j = 0; i < 10 ; i = i + 1) {} # j undefined
}
```

fail-for2.bawk

```
BEGIN {}

LOOP {}

END {
  int i;

  for (i = 0; j < 10 ; i = i + 1) {} # j undefined
}
```

fail-for3.bawk

```
BEGIN {}

LOOP {}

END {
  int i;

  for (i = 0; i ; i = i + 1) {} # i is an integer, not Boolean
}
```

fail-for4.bawk

```
BEGIN {}

LOOP {}

END {
  int i;

  for (i = 0; i < 10 ; i = j + 1) {} # j undefined
}
```

fail-for5.bawk

```
BEGIN {}

LOOP {}

END {
  int i;

  for (i = 0; i < 10 ; i = i + 1) {
    foo(); # Error: no function foo
  }
}
```

fail-for6.bawk

```
BEGIN {}

LOOP {}

END {
  int i;
  int x;

  x = 10;
  for (i in x) {} # Fail: enhanced for loop on an integer
}
```

fail-for7.bawk

```
BEGIN {}

LOOP {}

END {
  int x;
  string[] a;
  for(x in a) {} # Fail: Enhanced for loop on a string array using an int
}
```

fail-func1.bawk

```
BEGIN {
  function int foo() {}

  function int bar() {}

  function int baz() {}

  function void bar() {} # Error: duplicate function bar
}

LOOP {}

END {}
```

fail-func2.bawk

```
BEGIN {
  function int foo(int a, bool b, int c) { }

  function void bar(int a, bool b, int a) {} # Error: duplicate formal a in bar
}

LOOP {}

END {}
```

fail-func3.bawk

```
BEGIN {
  function int foo(int a, bool b, int c) { }

  function void bar(int a, void b, int c) {} # Error: illegal void formal b
}

LOOP {}

END {}
```

fail-func4.bawk

```
BEGIN {
  function int foo() {}

  function void bar() {}

  function int print() {} # Should not be able to define print

  function void baz() {}
}

LOOP {}

END {}
```

fail-func5.bawk

```
BEGIN {
  function int foo() {}

  function int bar() {
    int a;
    void b; # Error: illegal void local b
    bool c;

    return 0;
  }
}

LOOP {}

END {}
```

fail-func6.bawk

```
BEGIN {
  function void foo(int a, bool b)
  {
  }
}

LOOP {}

END {
  foo(42, true);
  foo(42); # Wrong number of arguments
}
```

fail-func7.bawk

```
BEGIN {
  function void foo(int a, bool b)
  {
  }
}

LOOP {}

END {
  foo(42, true);
  foo(42, true, false); # Wrong number of arguments
}
```

fail-func8.bawk

```
BEGIN {
  function void foo(int a, bool b)
  {
  }

  function void bar()
  {
  }
}

LOOP {}

END {
  foo(42, true);
  foo(42, bar()); # int and void, not int and bool
}
```

fail-func9.bawk

```
BEGIN {
  function void foo(int a, bool b)
  {
  }
}

LOOP {}

END {
  foo(42, true);
  foo(42, 42); # Fail: int, not bool
}
```

fail-global1.bawk

```
BEGIN {
  int c;
  bool b;
  void a; # global variables should not be void
}
```

```

LOOP {}

END {}
```

fail-global2.bawk

```
BEGIN {
  int b;
  bool c;
  int a;
  int b; # Duplicate global variable
}

LOOP {}

END {}
```

fail-helloworldbegin.bawk

```
BEGIN {
  print("Hello World!");
}

LOOP {}

END {
}

CONFIG {}
```

fail-if1.bawk

```
BEGIN {}

LOOP {}

END {
  if (true) {}
  if (false) {} else {}
  if (42) {} # Error: non-bool predicate
}
```

fail-if2.bawk

```
BEGIN {}

LOOP {}

END {
  if (true) {
    foo; # Error: undeclared variable
  }
}
```

fail-if3.bawk

```
BEGIN {}

LOOP {}

END {
  if (true) {
    42;
  } else {
    bar; # Error: undeclared variable
  }
}
```

fail-length.bawk

```
BEGIN {}

LOOP {}

END {
  int x;
  x = 0;
  length(x); # Fail: Finding length of int
}
```

fail-print1.bawk

```
BEGIN {
  # Should be illegal to redefine
  function void print() {}
}

LOOP {}

END {}
```

fail-print2.bawk

```
BEGIN {}

LOOP {}

END {
  print(5); # Fail: Printing an int instead of a string
}
```

fail-return1.bawk

```
BEGIN {
  function void foo()
  {
    if (true) {
      return 42; # Should return void
    }
    else {
      return;
    }
  }
}

LOOP {}

END {}
```

fail-scope1.bawk

```
BEGIN{}
LOOP{
  int i;

  {
    i = 15;
  }

  i = 32;
}
END{}
```

fail-structure1.bawk

```
BEGIN {
  function void loop() {}
}

LOOP {}

END {}
```

fail-structure2.bawk

```
BEGIN {
  function void end() {}
}

LOOP {}

END {}
```

fail-while1.bawk

```
BEGIN {}

LOOP {}

END {
  int i;

  while (true) {
    i = i + 1;
  }

  while (42) { # Should be boolean
    i = i + 1;
  }
}
```

fail-while2.bawk

```
BEGIN {}

LOOP {}

END {
  int i;

  while (true) {
    i = i + 1;
  }

  while (true) {
    foo(); # foo undefined
  }
}
```

pass-add1.bawk

```
BEGIN {
  function int add(int x, int y) {
    return x + y;
  }
}

LOOP {}

END {
  print(int_to_string(add(1, 2)));
}

CONFIG {}
```

pass-add2.bawk

```
BEGIN {
  function void add(int x, int y) {
    print(int_to_string(x + y));
  }
}

LOOP {}

END {
  add(1, 2);
}
```

pass-arith1.bawk

```
BEGIN {}

LOOP {}

END {
  int x;
  x = 0;
  x = x + 2;
  print(int_to_string(x));
}
```

pass-arith2.bawk

```
BEGIN {}

LOOP {}

END {
  print(int_to_string(1 + 2 * 3 + 4));
}
```

pass-arith3.bawk

```
BEGIN {}

LOOP {}

END {
  print(int_to_string(1 - 2 / 3 - 4));
}
```

pass-arith4.bawk

```
BEGIN {}

LOOP {}
```

```

END {
  int x;
  x = 2;
  x += 2;
  print(int_to_string(x));
}
```

pass-arith5.bawk

```
BEGIN {}

LOOP {}

END {
  int x;
  x = 2;
  x -= 2;
  print(int_to_string(x));
}
```

pass-arrayliterals.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1+1, 2+2, 3+2];
  print(int_to_string(a[0]));
}
```

pass-bool1.bawk

```
BEGIN {}

LOOP {}

END {
  bool b;
  b = true;
  if (b) {
    print(int_to_string(10));
  }
}
```

pass-bool2.bawk

```
BEGIN {
  function bool cond() {
    bool b;
    b = true;
    return b;
  }
}

LOOP {}

END {
  if (cond()) {
    print(int_to_string(10));
  }
}
```

pass-boolarr1.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [];
  print(int_to_string(length(a)));
}
```

pass-boolarr2.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  print(int_to_string(length(a)));
}
```

pass-boolarr3.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  print(bool_to_string(a[0]));
}
```

pass-boolarr4.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  a[0] = false;
  print(bool_to_string(a[0]));
}
```

pass-boolarr5.bawk

```
BEGIN {
  function bool[] foo() {
    bool[] a;
    a = [true, false, false];
    return a;
  }
}

LOOP {}

END {
  bool[] a;
  a = foo();
  print(bool_to_string(a[0]));
}
```

pass-boolarr6.bawk

```
BEGIN {
  function bool[] foo (bool[] a) {
    return a;
  }
}

LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  a = foo(a);
  print(bool_to_string(a[0]));
}
```

pass-boolarr7.bawk

```
BEGIN {}

LOOP {}
```

```

END {
  bool[][] m;
  bool[][][] a;
  bool[][][][] b;

  m = [[false, false], [false, false]];
  m[0][0] = true;
  print(bool_to_string(m[0][0]));

  a =
  [
    [ [true, false], [false, false] ], [ [false, false], [false, false] ]
  ];
  print(bool_to_string(a[0][0][0]));

  b =
  [
    [ [ [true, false], [false, false] ], [ [false, false], [false, false] ] ],
    [ [ [false, false], [false, false] ], [ [false, false], [false, false] ] ]
  ];
  print(bool_to_string(b[0][0][0][0]));
}
```

pass-boolarr8.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [false, false, false];
  print(bool_to_string(contains(a, false)));
  print(bool_to_string(contains(a, true)));
}
```

pass-boolarr9.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  print(int_to_string(index_of(a, true)));
}
```

pass-comment.bawk

```
BEGIN {
  function int add(int x, int y) {
    return x + y;
  }
}

LOOP {}

END {
  # this should not print
  # print(int_to_string(add(1, 2)));
  print(int_to_string(add(1, 2)));
}

CONFIG {}
```

pass-config1.bawk

```
BEGIN {}

LOOP {
  print($1);
}

END {}

CONFIG {
  RS = "\n";
}
```

pass-config2.bawk

```
BEGIN {}

LOOP {
  print($1);
}

END {}

CONFIG {
  FS = " ";
}
```

pass-config3.bawk

```
BEGIN {}

LOOP {
  print($1);
}

END {}
```

```

CONFIG {
  RS = "\n";
  FS = "  ";
}
```

pass-dollar1.bawk

```
BEGIN {}

LOOP {
  print($0);
}

END {}
```

pass-dollar2.bawk

```
BEGIN {}

LOOP {
  string hello;
  hello = $0;
  print(hello);
}

END {}
```

pass-dynamicarr1.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  insert(a, 3, true);
  print(int_to_string(length(a)));
}
```

pass-dynamicarr2.bawk

```
BEGIN {}

LOOP {}

END {
  bool[] a;
  a = [true, false, false];
  delete(a, 2);
  print(int_to_string(length(a)));
}
```

pass-dynamicarr3.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  int b;
  a = [1,2,3];
  b = a[3]; # Fail: Accessing past size of array
}
```

pass-dynamicarr4.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1,2,3];
  insert(a, 6, 4); # Fail: Inserting past (size of array + 1)
}
```

pass-dynamicarr5.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1,2,3];
  insert(a, -1, 4); # Fail: Inserting at illegal index
}
```

pass-dynamicarr6.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1,2,3];
  delete(a, 4); # Fail: Deleting element from invalid index
}
```

pass-dynamicarr7.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1,2,3];
  delete(a, -1); # Fail: Deleting from an invalid index
}
```

pass-fib.bawk

```
BEGIN {
  function int fib(int x)
  {
    if (x < 2) {
      return 1;
    }
    return fib(x-1) + fib(x-2);
  }
}

LOOP {}

END {
  print(int_to_string(fib(0)));
  print(int_to_string(fib(1)));
  print(int_to_string(fib(2)));
  print(int_to_string(fib(3)));
  print(int_to_string(fib(4)));
  print(int_to_string(fib(5)));
}
```

pass-for1.bawk

```
BEGIN {}

LOOP {}

END {
  int i;
  for (i = 0 ; i < 5 ; i = i + 1) {
    print(int_to_string(i));
  }
  print(int_to_string(42));
}
```

pass-for2.bawk

```
BEGIN {}

```

```
LOOP {}

END {
  int i;
  int[] arr;
  arr = [1, 2, 3];

  for (i = 0; i < length(arr); i++) {
    print(int_to_string(arr[i]));
  }
  print(int_to_string(42));
}
```

pass-func1.bawk

```
BEGIN {
  function int fun(int x, int y)
  {
    return 0;
  }
}

LOOP {}

END {
  int i;
  i = 1;

  fun(i = 2, i = i+1);

  print(int_to_string(i));
}
```

pass-func2.bawk

```
BEGIN {
  function void printem(int a, int b, int c, int d)
  {
    print(int_to_string(a));
    print(int_to_string(b));
    print(int_to_string(c));
    print(int_to_string(d));
  }
}

LOOP {}

END {
  printem(42,17,192,8);
}
```

pass-func3.bawk

```
BEGIN {
  function int add(int a, int b)
  {
    int c;
    c = a + b;
    return c;
  }
}

LOOP {}

END {
  int d;
  d = add(52, 10);
  print(int_to_string(d));
}
```

pass-func4.bawk

```
BEGIN {
  function int foo(int a)
  {
    return a;
  }
}

LOOP {}

END {
  print( int_to_string(foo(2)) );
}
```

pass-func5.bawk

```
BEGIN {
  function void foo() {}

  function int bar(int a, bool b, int c) { return a + c; }
}

LOOP {}

END {
  print(int_to_string(bar(17, false, 25)));
}
```

pass-func6.bawk

```
BEGIN {
  int a;

  function void foo(int c)
  {
    a = c + 42;
  }
}

LOOP {}

END {
  foo(73);
  print(int_to_string(a));
}
```

pass-func7.bawk

```
BEGIN {
  function void foo(int a)
  {
    print(int_to_string(a + 3));
  }
}

LOOP {}

END {
  foo(40);
}
```

pass-func8.bawk

```
BEGIN {
  function void foo(int a)
  {
    print(int_to_string(a + 3));
    return;
  }
}

LOOP {}

END {
  foo(40);
}
```

pass-func9.bawk

```
BEGIN {
  function int x() {
    return 2;
  }

  function int y() {
    return x();
  }
}

LOOP {}

END {
  print( int_to_string(y()) );
}
```

pass-gcd.bawk

```
BEGIN {
  function int gcd(int a, int b) {
    while (a != b) {
      if (a > b) {
        a = a - b;
      }
      else {
        b = b - a;
      }
    }
    return a;
  }
}

LOOP {}

END {
  print(int_to_string(gcd(2,14)));
  print(int_to_string(gcd(3,15)));
  print(int_to_string(gcd(99,121)));
}
```

pass-global1.bawk

```
BEGIN {
  int a;
  int b;

  function int add(int x, int y) {
    return x + y;
  }
}

LOOP {}

END {
  a = 1;
  b = 2;
  print(int_to_string(add(a, b)));
}
```

pass-global2.bawk

```
BEGIN {
  int a;
  int b;

  function void printa()
  {
    print(int_to_string(a));
  }

  function void printbb()
  {
    print(int_to_string(b));
  }

  function void incab()
  {
    a = a + 1;
    b = b + 1;
  }
}

LOOP {}

END {
  a = 42;
  b = 21;
  printa();
  printbb();
  incab();
  printa();
  printbb();
}
```

pass-global3.bawk

```
BEGIN {
  bool i;
}

LOOP {}

END {
  int i; # Should hide the global i

  i = 42;
  print(int_to_string(i + i));
}
```

pass-helloworld.bawk

```
BEGIN {}

```

```
LOOP {}

END {
  print("Hello World!");
}

CONFIG {}
```

pass-helloworldloop.bawk

```
BEGIN {}

LOOP {
  print("Hello World");
}

END {}

CONFIG {}
```

pass-helloworldloopend.bawk

```
BEGIN {}

LOOP {
  print("Hello loop!");
}

END {
  print("Hello end!");
}
```

pass-if1.bawk

```
BEGIN {}

LOOP {}

END {
  if (true) {
    print(int_to_string(10));
  }
  print(int_to_string(2));
}
```

pass-if2.bawk

```
BEGIN {}

LOOP {}

```

```
END {
  if(true) {
    print(int_to_string(10));
  }
  else {
    print(int_to_string(5));
  }
  print(int_to_string(2));
}
```

pass-if3.bawk

```
BEGIN {}

LOOP {}

END {
  if(false) {
    print(int_to_string(10));
  }
  print(int_to_string(2));
}
```

pass-if4.bawk

```
BEGIN {}

LOOP {}

END {
  if(false) {
    print(int_to_string(10));
  }
  else {
    print(int_to_string(5));
  }
  print(int_to_string(2));
}
```

pass-if5.bawk

```
BEGIN {
  function int cond(bool b) {
    int x;
    if (b) {
      x = 10;
    }
    else {
      x = 5;
    }
    return x;
  }
}

LOOP {}

END {
  print(int_to_string(cond(true)));
  print(int_to_string(cond(false)));
}
```

pass-if6.bawk

```
BEGIN {
  function int cond(bool b) {
    int x;
    if (!b) {
      x = 10;
    }
    else {
      x = 5;
    }
    return x;
  }
}

LOOP {}

END {
  print(int_to_string(cond(true)));
  print(int_to_string(cond(false)));
}
```

pass-if7.bawk

```
BEGIN {
  function int cond(bool b) {
    int x;
    x = 10;
    if (b) {
      if (x == 10) {
        x = 10;
      }
    }
    else {
      x = 5;
    }
    return x;
  }
}

LOOP {}

END {
  print(int_to_string(cond(true)));
  print(int_to_string(cond(false)));
}
```

pass-intarr1.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [];
  print(int_to_string(length(a)));
}
```

pass-intarr2.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1, 2, 3];
  print(int_to_string(length(a)));
}
```

pass-intarr3.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1, 2, 3];
  print(int_to_string(a[0]));
}
```

pass-intarr4.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1, 2, 3];
  a[0] = 0;
  print(int_to_string(a[0]));
}
```

pass-intarr5.bawk

```
BEGIN {
  function int[] foo() {
    int[] a;
    a = [1, 2, 3];
    return a;
  }
}

LOOP {}

END {
  int[] a;
  a = foo();
  print(int_to_string(a[0]));
}
```

pass-intarr6.bawk

```
BEGIN {
  function int[] foo (int[] a) {
    return a;
  }
}

LOOP {}

END {
  int[] a;
  a = [1, 2, 3];
  a = foo(a);
  print(int_to_string(a[0]));
}
```

pass-intarr7.bawk

```
BEGIN {}

LOOP {}

END {
  int[][] m;
  int[][][] a;
  int[][][][] b;

  m = [[1, 2], [3, 4]];
  m[0][0] = 20;
  print(int_to_string(m[0][0]));

  a =
  [
    [ [1, 2], [3, 4] ], [ [5, 6], [7, 8] ]
  ];
  print(int_to_string(a[0][0][0]));

  b =
  [
    [ [ [1, 2], [3, 4] ], [ [5, 6], [7, 8] ] ],
    [ [ [9, 10], [11, 12] ], [ [13, 14], [15, 16] ] ]
  ];
  print(int_to_string(b[0][0][0][0]));
}
```

pass-intarr8.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1, 2, 3];
  print(bool_to_string(contains(a, 1)));
  print(bool_to_string(contains(a, 4)));
}
```

pass-intarr9.bawk

```
BEGIN {}

LOOP {}

END {
  int[] a;
  a = [1, 2, 3];
  print(int_to_string(index_of(a, 1)));
}
```

pass-local1.bawk

```
BEGIN {
  function void foo(bool i)
  {
    int i; # Should hide the formal i

    i = 42;
    print(int_to_string(i + i));
  }
}

LOOP {}

END {
  foo(true);
}
```

pass-local2.bawk

```
BEGIN {
  function int foo(int a, bool b)
  {
    int c;
    bool d;

    c = a;

    return c + 10;
  }
}

LOOP {}

END {
  print(int_to_string(foo(37, false)));
}
```

pass-nf1.bawk

```
BEGIN {}

LOOP {
  print(int_to_string(NF));
}

END {}
```

pass-ops1.bawk

```
BEGIN {}

LOOP {}

END {
  print(int_to_string(1 + 2));
  print(int_to_string(1 - 2));
  print(int_to_string(-1 + 2));
  print(int_to_string(1 * 2));
  print(int_to_string(100 / 2));
  print(int_to_string(99));
  print(bool_to_string(1 == 2));
  print(bool_to_string(1 == 1));
  print(int_to_string(99));
  print(bool_to_string(1 != 2));
  print(bool_to_string(1 != 1));
  print(int_to_string(99));
  print(bool_to_string(1 < 2));
  print(bool_to_string(2 < 1));
  print(int_to_string(99));
  print(bool_to_string(1 <= 2));
  print(bool_to_string(1 <= 1));
  print(bool_to_string(2 <= 1));
  print(int_to_string(99));
  print(bool_to_string(1 > 2));
  print(bool_to_string(2 > 1));
  print(int_to_string(99));
  print(bool_to_string(1 >= 2));
  print(bool_to_string(1 >= 1));
  print(bool_to_string(2 >= 1));
}
```

pass-ops2.bawk

```
BEGIN {}

LOOP {}

END {
  int x;
  x = 42;
  print(bool_to_string(true));
  print(bool_to_string(false));
  print(bool_to_string(true && true));
  print(bool_to_string(true && false));
  print(bool_to_string(false && true));
  print(bool_to_string(false && false));
  print(bool_to_string(true || true));
  print(bool_to_string(true || false));
  print(bool_to_string(false || true));
  print(bool_to_string(false || false));
  print(bool_to_string(!false));
  print(bool_to_string(!true));
  print(int_to_string(-10));
  print(int_to_string(x--));
  print(int_to_string(x++));
}
```

pass-ops3.bawk

```
BEGIN {}

LOOP {}

END {
  print("hello " & "world");
  print(int_to_string(99));
  print(bool_to_string("hello" == "world"));
  print(bool_to_string("hello" == "hello"));
  print(int_to_string(99));
  print(bool_to_string("hello" != "world"));
  print(bool_to_string("hello" != "hello"));
  print(int_to_string(99));
  print(bool_to_string("hello" < "world"));
  print(bool_to_string("world" < "hello"));
  print(int_to_string(99));
  print(bool_to_string("hello" <= "world"));
  print(bool_to_string("hello" <= "hello"));
  print(bool_to_string("world" <= "hello"));
  print(int_to_string(99));
  print(bool_to_string("hello" > "world"));
  print(bool_to_string("world" > "hello"));
  print(int_to_string(99));
  print(bool_to_string("hello" >= "world"));
  print(bool_to_string("hello" >= "hello"));
  print(bool_to_string("world" >= "hello"));
}
```

pass-ops4.bawk

```
BEGIN {}

LOOP {}

END {
  print(bool_to_string('[0-9]*' % '[0]*'));
  print(bool_to_string('[0-9]*' % '[0-9]*'));
  print(int_to_string(99));
  print(bool_to_string('[0-9]*' !% '[0]*'));
  print(bool_to_string('[0-9]*' !% '[0-9]*'));
  print(int_to_string(99));
  print(bool_to_string("1234" ~ '[0]'));
  print(bool_to_string("1234" ~ '[0-9]*'));
  print(int_to_string(99));
  print(bool_to_string("1234" !~ '[0]'));
  print(bool_to_string("1234" !~ '[0-9]*'));
}
```

pass-print.bawk

```
BEGIN {}

LOOP {
  print(int_to_string(string_to_int("8")));
}

END {}
```

pass-printbegin.bawk

```
BEGIN {
  function void print_begin () {
    print("Hello world");
  }
}
LOOP {}
END {
  print_begin();
}
```

pass-rgx1.bawk

```
BEGIN {}

LOOP {}

END {
  rgx r;
  r = '[0-9]*';
  print(rgx_to_string(r));
}
```

pass-rgx2.bawk

```
BEGIN {
  function rgx foo() {
    return '[0-9]*';
  }
}

LOOP {}

END {
  print(rgx_to_string(foo()));
}

CONFIG {}
```

pass-rgx3.bawk

```
BEGIN {
  function bool foo(rgx a, rgx b) {
    return a % b;
  }
}

LOOP {}

END {
  print(bool_to_string(foo('[0-9]*', '[0-9]*')));
}

CONFIG {}
```

pass-rgx4.bawk

```
BEGIN {
  function bool foo(rgx a, rgx b) {
    return a !% b;
  }
}

LOOP {}

END {
  print(bool_to_string(foo('[0-9]*', '[0-9]*')));
}

CONFIG {}
```

pass-rgx5.bawk

```
BEGIN {
  function bool foo(string a, rgx b) {
    return a ~ b;
  }
}

LOOP {}

END {
  print(bool_to_string(foo("1234", '[0-9]*')));
}

CONFIG {}
```

pass-rgx6.bawk

```
BEGIN {
  function bool foo(string a, rgx b) {
    return a !~ b;
  }
}

LOOP {}

END {
  print(bool_to_string(foo("1234", '[0]')));
}

CONFIG {}
```

pass-rgxarr1.bawk

```
BEGIN {}

LOOP {}

END {
  rgx[] a;
  a = [];
  print(int_to_string(length(a)));
}
```

pass-rgxarr2.bawk

```
BEGIN {}

LOOP {}

END {
  rgx[] a;
  a = ['[0]*', '[1]*', '[1]*'];
  print(int_to_string(length(a)));
}
```

pass-rgxarr3.bawk

```
BEGIN {}

LOOP {}

END {
  rgx[] a;
  a = ['[0]*', '[1]*', '[1]*'];
  print(rgx_to_string(a[0]));
}
```

pass-rgxarr4.bawk

```
BEGIN {}

LOOP {}

END {
  rgx[] a;
  a = ['[0]*', '[1]*', '[1]*'];
  a[0] = '[2]*';
  print(rgx_to_string(a[0]));
}
```

pass-rgxarr5.bawk

```
BEGIN {
  function rgx[] foo() {
    rgx[] a;
    a = ['[0]*', '[1]*', '[1]*'];
    return a;
  }
}

LOOP {}
```

```

END {
  rgx[] a;
  a = foo();
  print(rgx_to_string(a[0]));
}
```

pass-rgxarr6.bawk

```
BEGIN {
  function rgx[] foo (rgx[] a) {
    return a;
  }
}

LOOP {}

END {
  rgx[] a;
  a = ['[0]*', '[1]*', '[1]*'];
  a = foo(a);
  print(rgx_to_string(a[0]));
}
```

pass-rgxarr7.bawk

```
BEGIN {}

LOOP {}

END {
  rgx[][] m;
  rgx[][][] a;
  rgx[][][][] b;

  m = [['[0*]', '[1*]'], ['[1*]', '[1*]']];
  m[0][0] = '[6*]';
  print(rgx_to_string(m[0][0]));

  a =
  [
    [ ['[0*]', '[1*]'], ['[1*]', '[1*]'] ], [ ['[1*]', '[1*]'], ['[1*]', '[1*]'] ]
  ];
  print(rgx_to_string(a[0][0][0]));

  b =
  [
    [ [ ['[0*]', '[1*]'], ['[1*]', '[1*]'] ], [ ['[1*]', '[1*]'], ['[1*]', '[1*]']
        ] ],
    [ [ ['[1*]', '[1*]'], ['[1*]', '[1*]'] ], [ ['[1*]', '[1*]'], ['[1*]', '[1*]']
        ] ]
  ];
  print(rgx_to_string(b[0][0][0][0]));
}
```

pass-rgxarr8.bawk

```
BEGIN {}

LOOP {}

END {
  rgx[] a;
  a = ['[0]*', '[1]*', '[1]*'];
  print(bool_to_string(contains(a, '[0]*')));
  print(bool_to_string(contains(a, '[2]*')));
}
```

pass-rgxarr9.bawk

```
BEGIN {}

LOOP {}

END {
  rgx[] a;
  a = ['[0]*', '[1]*', '[1]*'];
  print(int_to_string(index_of(a, '[0]*')));
}
```

pass-strarr1.bawk

```
BEGIN {}

LOOP {}

END {
  string[] a;
  a = [];
  print(int_to_string(length(a)));
}
```

pass-strarr2.bawk

```
BEGIN {}

LOOP {}

END {
  string[] a;
  a = ["a", "b", "c"];
  print(int_to_string(length(a)));
}
```

pass-strarr3.bawk

```
BEGIN {}

LOOP {}

END {
  string[] a;
  a = ["a", "b", "c"];
  print(a[0]);
}
```

pass-strarr4.bawk

```
BEGIN {}

LOOP {}

END {
  string[] a;
  a = ["a", "b", "c"];
  a[0] = "p";
  print(a[0]);
}
```

pass-strarr5.bawk

```
BEGIN {
  function string[] foo() {
    string[] a;
    a = ["a", "b", "c"];
    return a;
  }
}

LOOP {}

END {
  string[] a;
  a = foo();
  print(a[0]);
}
```

pass-strarr6.bawk

```
BEGIN {
  function string[] foo (string[] a) {
    return a;
  }
}

LOOP {}

```

```
END {
  string[] a;
  a = ["a", "b", "c"];
  a = foo(a);
  print(a[0]);
}
```

pass-strarr7.bawk

```
BEGIN {}

LOOP {}

END {
  string[][] m;
  string[][][] a;
  string[][][][] b;

  m = [["a", "b"], ["c", "d"]];
  m[0][0] = "hi";
  print(m[0][0]);

  a =
  [
    [ ["a", "b"], ["c", "d"] ], [ ["e", "f"], ["g", "h"] ]
  ];
  print(a[0][0][0]);

  b =
  [
    [ [ ["a", "b"], ["c", "d"] ], [ ["e", "f"], ["g", "h"] ] ],
    [ [ ["i", "j"], ["k", "l"] ], [ ["m", "n"], ["o", "p"] ] ]
  ];
  print(b[0][0][0][0]);
}
```

pass-strarr8.bawk

```
BEGIN {}

LOOP {}

END {
  string[] a;
  a = ["a", "b", "c"];
  print(bool_to_string(contains(a, "a")));
  print(bool_to_string(contains(a, "e")));
}
```

pass-strarr9.bawk

```
BEGIN {}
```

```

LOOP {}

END {
  string[] a;
  a = ["a", "b", "c"];
  print(int_to_string(index_of(a, "a")));
}
```

pass-string1.bawk

```
BEGIN {}

LOOP {}

END {
  string s;
  s = "hello";
  print(s);
}
```

pass-string2.bawk

```
BEGIN {
  function string foo() {
    return "hello";
  }
}

LOOP {}

END {
  print(foo());
}

CONFIG {}
```

pass-string3.bawk

```
BEGIN {
  function string concat(string a, string b) {
    return a & b;
  }
}

LOOP {}

END {
  print(concat("hello ", "world"));
}

CONFIG {}
```

pass-string4.bawk

```
BEGIN {}

LOOP {}

END {
  int x;
  string s;

  s = "3";
  x = string_to_int(s);
  print(int_to_string(x));
}
```

pass-while1.bawk

```
BEGIN {}

LOOP {}

END {
  int i;
  i = 5;
  while (i > 0) {
    print(int_to_string(i));
    i = i - 1;
  }
  print(int_to_string(42));
}
```

pass-while2.bawk

```
BEGIN {
  function int foo(int a)
  {
    int j;
    j = 0;
    while (a > 0) {
      j = j + 2;
      a = a - 1;
    }
    return j;
  }
}

LOOP {}

END {
  print(int_to_string(foo(7)));
}
```

Failed tests output

fail-FS.err

```
Fatal error: exception Parsing.Parse_error
```

fail-RS.err

```
Fatal error: exception Parsing.Parse_error
```

fail-array1.err

```
Fatal error: exception Parsing.Parse_error
```

fail-array2.err

```
Fatal error: exception Failure("cannot perform contains on int[] and bool")
```

fail-array3.err

```
Fatal error: exception Failure("cannot perform index_of on int[] and bool")
```

fail-arrayassign1.err

```
Fatal error: exception Failure("illegal assignment int[] = int[][] in a = [[1, 2],
     [3, 4]]")
```

fail-arrayassign2.err

```
Fatal error: exception Failure("array of different types, expected int found
    string")
```

fail-arrayassign3.err

```
Fatal error: exception Failure("array of different types, expected int[] found
    string[]")
```

fail-assign1.err

```
Fatal error: exception Failure("illegal assignment int = bool in i = false")
```

fail-assign2.err

```
Fatal error: exception Failure("illegal assignment bool = int in b = 48")
```

fail-assign3.err

```
Fatal error: exception Failure("illegal assignment int = void in i = myvoid()")
```

fail-assign4.err

```
Fatal error: exception Failure("illegal void local x")
```

fail-assign5.err

```
Fatal error: exception Failure("illegal assignment string = rgx in s = '[0]*'")
```

fail-assign6.err

```
Fatal error: exception Failure("illegal assignment of NF")
```

fail-assign7.err

```
Fatal error: exception Failure("illegal assignment rgx = string in r = hello")
```

fail-blocks1.err

```
Fatal error: exception Parsing.Parse_error
```

fail-conversion1.err

```
Fatal error: exception Failure("illegal argument found int expected string in 8")
```

fail-conversion2.err

```
Fatal error: exception Failure("illegal argument found string expected int in hi")
```

fail-conversion3.err

```
Fatal error: exception Failure("illegal argument found int expected rgx in 2")
```

fail-conversion4.err

```
Fatal error: exception Failure("illegal argument found rgx expected bool in
    '[0-9]*'")
```

fail-conversion5.err

```
Fatal error: exception Failure("illegal argument found rgx expected bool in
    '[0-9]*'")
```

fail-dead1.err

```
Fatal error: exception Failure("nothing may follow a return")
```

fail-decl1.err

```
Fatal error: exception Parsing.Parse_error
```

fail-decl2.err

```
Fatal error: exception Parsing.Parse_error
```

fail-decl3.err

```
Fatal error: exception Parsing.Parse_error
```

fail-dollarbegin.err

```
Fatal error: exception Parsing.Parse_error
```

fail-dynamicarr1.err

```
Fatal error: exception Failure("cannot perform insert on int[] and bool")
```

fail-expr1.err

```
Fatal error: exception Failure("illegal assignment int = bool in i = false")
```

fail-expr2.err

```
Fatal error: exception Failure("illegal binary operator bool + int in d + a")
```

fail-expr3.err

```
Fatal error: exception Failure("illegal binary operator bool + int in b + a")
```

fail-expr4.err

```
Fatal error: exception Failure("illegal binary operator string + int in b + a")
```

fail-expr5.err

```
Fatal error: exception Failure("Different types expected for '[0]*'~true")
```

fail-for1.err

```
Fatal error: exception Failure("undeclared identifier j")
```

fail-for2.err

```
Fatal error: exception Failure("undeclared identifier j")
```

fail-for3.err

```
Fatal error: exception Failure("expected Boolean expression in i")
```

fail-for4.err

```
Fatal error: exception Failure("undeclared identifier j")
```

fail-for5.err

```
Fatal error: exception Failure("unrecognized function foo")
```

fail-for6.err

```
Fatal error: exception Failure("cannot iterate over type int")
```

fail-for7.err

```
Fatal error: exception Failure("mismatch in int and string[]")
```

fail-func1.err

```
Fatal error: exception Failure("duplicate function bar")
```

fail-func2.err

```
Fatal error: exception Failure("duplicate formal a")
```

fail-func3.err

```
Fatal error: exception Failure("illegal void formal b")
```

fail-func4.err

```
Fatal error: exception Failure("function print may not be defined")
```

fail-func5.err

```
Fatal error: exception Failure("illegal void local b")
```

fail-func6.err

```
Fatal error: exception Failure("expecting 2 arguments in foo(42)")
```

fail-func7.err

```
Fatal error: exception Failure("expecting 2 arguments in foo(42, true, false)")
```

fail-func8.err

```
Fatal error: exception Failure("illegal argument found void expected bool in bar()
    ")
```

fail-func9.err

```
Fatal error: exception Failure("illegal argument found int expected bool in 42")
```

fail-global1.err

```
Fatal error: exception Failure("illegal void global a")
```

fail-global2.err

```
Fatal error: exception Failure("duplicate global b")
```

fail-helloworldbegin.err

```
Fatal error: exception Parsing.Parse_error
```

fail-if1.err

```
Fatal error: exception Failure("expected Boolean expression in 42")
```

fail-if2.err

```
Fatal error: exception Failure("undeclared identifier foo")
```

fail-if3.err

```
Fatal error: exception Failure("undeclared identifier bar")
```

fail-length.err

```
Fatal error: exception Failure("illegal argument found int arraytype expected in
    length(x)")
```

fail-print1.err

```
Fatal error: exception Failure("function print may not be defined")
```

fail-print2.err

```
Fatal error: exception Failure("illegal argument found int expected string in 5")
```

fail-return1.err

```
Fatal error: exception Failure("return gives int expected void in 42")
```

fail-scope1.err

```
Fatal error: exception Parsing.Parse_error
```

fail-structure1.err

```
Fatal error: exception Failure("function loop may not be defined")
```

fail-structure2.err

```
Fatal error: exception Failure("function end may not be defined")
```

fail-while1.err

```
Fatal error: exception Failure("expected Boolean expression in 42")
```

fail-while2.err

```
Fatal error: exception Failure("unrecognized function foo")
```

Passed tests output `pass-add1.out`

```
3
```

`pass-add2.out`

```
3
```

`pass-arith1.out`

```
2
```

`pass-arith2.out`

```
11
```

`pass-arith3.out`

```
-3
```

`pass-arith4.out`

```
4
```

`pass-arith5.out`

```
0
```

`pass-arrayliterals.out`

```
2
```

`pass-bool1.out`

```
10
```

`pass-bool2.out`

```
10
```

`pass-boolarr1.out`

```
0
```

pass-boolarr2.out

```
3
```

pass-boolarr3.out

```
true
```

pass-boolarr4.out

```
false
```

pass-boolarr5.out

```
true
```

pass-boolarr6.out

```
true
```

pass-boolarr7.out

```
true
true
true
```

pass-boolarr8.out

```
true
false
```

pass-boolarr9.out

```
0
```

pass-comment.out

```
3
```

pass-config1.out

```
Hello
Line
This
Wow
Here
```

pass-config2.out

```
Hello
Line
This
Wow
Here
```

pass-config3.out

```
Hello
Line
This
Wow
Here
```

pass-dollar1.out

```
Hello
Line 2
This is the best line
Wow
Here we go
```

pass-dollar2.out

```
Hello
Line 2
This is the best line
Wow
Here we go
```

pass-dynamicarr1.out

```
4
```

pass-dynamicarr2.out

```
2
```

pass-dynamicarr3.out

```
Index Out of Bounds Error.
```

pass-dynamicarr4.out

```
Index Out of Bounds Error.
```

pass-dynamicarr5.out

```
Index Out of Bounds Error.
```

pass-dynamicarr6.out

```
Index Out of Bounds Error.
```

pass-dynamicarr7.out

```
Index Out of Bounds Error.
```

pass-fib.out

```
1
1
2
3
5
8
```

pass-for1.out

```
0
1
2
3
4
42
```

pass-for2.out

```
1
2
3
42
```

pass-func1.out

```
2
```

pass-func2.out

```
42
17
192
8
```

pass-func3.out

```
62
```

pass-func4.out

```
2
```

pass-func5.out

```
42
```

pass-func6.out

```
115
```

pass-func7.out

```
43
```

pass-func8.out

```
43
```

pass-func9.out

```
2
```

pass-gcd.out

```
2
3
11
```

pass-global1.out

```
3
```

pass-global2.out

```
42
21
43
22
```

pass-global3.out

```
84
```

pass-helloworld.out

```
Hello World!
```

pass-helloworldloop.out

```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

pass-helloworldloopend.out

```
Hello loop!
Hello loop!
Hello loop!
Hello loop!
Hello loop!
Hello loop!
Hello end!
```

pass-if1.out

```
10
2
```

pass-if2.out

```
10
2
```

pass-if3.out

```
2
```

pass-if4.out

```
5
2
```

pass-if5.out

```
10
5
```

pass-if6.out

```
5
10
```

pass-if7.out

```
10
5
```

pass-intarr1.out

```
0
```

pass-intarr2.out

```
3
```

pass-intarr3.out

```
1
```

pass-intarr4.out

```
0
```

pass-intarr5.out

```
1
```

pass-intarr6.out

```
1
```

pass-intarr7.out

```
20
1
1
```

pass-intarr8.out

```
true
false
```

pass-intarr9.out

```
0
```

pass-local1.out

```
84
```

pass-local2.out

```
47
```

pass-nf1.out

```
1
2
5
1
3
0
```

pass-ops1.out

```
3
-1
1
2
50
99
false
true
99
true
false
99
true
false
99
true
true
false
99
false
true
99
```

```
false
true
true
```

pass-ops2.out

```
true
false
true
false
false
false
true
true
true
false
true
false
-10
41
42
```

pass-ops3.out

```
hello world
99
false
true
99
true
false
99
true
false
99
true
true
false
99
false
true
99
false
true
true
```

pass-ops4.out

```
false
true
99
true
```

```
false
99
false
true
99
true
false
```

pass-print.out

```
8
8
8
8
8
8
```

pass-printbegin.out

```
Hello world
```

pass-rgx1.out

```
[0-9]*
```

pass-rgx2.out

```
[0-9]*
```

pass-rgx3.out

```
true
```

pass-rgx4.out

```
false
```

pass-rgx5.out

```
true
```

pass-rgx6.out

```
true
```

pass-rgxarr1.out

```
0
```

pass-rgxarr2.out

```
3
```

pass-rgxarr3.out

```
[0]*
```

pass-rgxarr4.out

```
[2]*
```

pass-rgxarr5.out

```
[0]*
```

pass-rgxarr6.out

```
[0]*
```

pass-rgxarr7.out

```
[6*]
[0*]
[0*]
```

pass-rgxarr8.out

```
true
false
```

pass-rgxarr9.out

```
0
```

pass-strarr1.out

```
0
```

pass-strarr2.out

```
3
```

pass-strarr3.out

```
a
```

pass-strarr4.out

```
p
```

pass-strarr5.out

```
a
```

pass-strarr6.out

```
a
```

pass-strarr7.out

```
hi
a
a
```

pass-strarr8.out

```
true
false
```

pass-strarr9.out

```
0
```

pass-string1.out

```
hello
```

pass-string2.out

```
hello
```

pass-string3.out

```
hello world
```

pass-string4.out

```
3
```

pass-while1.out

```
5
4
3
2
1
42
```

pass-while2.out

```
14
```

Demo program For our demo, we put an ASCII art image in a file, split it by columns delimited by ”—”, and shuffled the columns. Then, we took an excerpt of text (from the Constitution) and injected it randomly into the shuffled columns. The bawk program that shuffled the original image is listed below:

shuffled.bawk

```
BEGIN {}
LOOP {
  nprint("|");
  nprint($5);
  nprint("|");
  nprint($3);
  nprint("|");
  nprint($1);
  nprint("|");
  nprint($2);
  nprint("|");
  nprint($4);
  nprint("|");
  nprint($6);
  nprint("|");
  nprint("\n");
}
END {}
CONFIG {
  FS = "|";
}
```

The goal of our demo was to unshuffle the file and use regex matching to remove and isolate the Constitution words. We also wanted to demonstrate the functionality of our multidimensional arrays, so we wrote a function to perform matrix multiplication and ran it on some inputs in the `END` block. The full text of the demo program is listed below.

`demo.bawk`

```
BEGIN {
  function int[][] matrix_mult(int[][] product, int[][] a, int[][] b) {
    int i;
    int j;
    int k;

    for (i = 0; i < length(a); i++) {
      for (j = 0; j < length(b[0]); j++) {
        for (k = 0; k < length(a[0]); k++) {
          product[i][j] += a[i][k] * b[k][j];
        }
      }
    }

    return product;
  }
}

LOOP {
  int i;
  int j;
  int col;
  int[] reshuffle_arr;
  string constitution_word;
  rgx rgx_match;

  reshuffle_arr = [3, 4, 2, 5, 1, 6];
  rgx_match = '\*[a-zA-Z]*\*';
  constitution_word = "";

  # Reshuffle columns
  for (i = 0; i < length(reshuffle_arr); i++) {
    col = reshuffle_arr[i];
    if ($col !~ rgx_match) {
      nprint($col);
    }
    else {
      constitution_word = $col;
    }
  }
  nprint("\t");
  nprint(constitution_word);
  nprint("\n");
}

END {
  int i;
  int j;
  int k;
  int l;
  int m;
  int n;
  int[][] a;
  int[][] b;
  int[][] product;

  a =       [[1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3]];
  b =       [[1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3],
             [1,2,3,1,2,3]];
  product = [[0,0,0,0,0,0],
             [0,0,0,0,0,0],
             [0,0,0,0,0,0],
             [0,0,0,0,0,0],
             [0,0,0,0,0,0],
             [0,0,0,0,0,0]];
  product = matrix_mult(product, a, b);

  # Print A
  print("A: ");
  for (i = 0; i < length(a); i++) {
    for (j = 0; j < length(a[0]); j++) {
      nprint(int_to_string(a[i][j]));
      nprint(" ");
    }
    nprint("\n");
  }
  nprint("\n");

  # Print B
  print("B: ");
  for (k = 0; k < length(b); k++) {
    for (l = 0; l < length(b[0]); l++) {
      nprint(int_to_string(b[k][l]));
      nprint(" ");
    }
    nprint("\n");
  }
  nprint("\n");

  # Print A x B
  print("A x B: ");
  for (m = 0; m < length(product); m++) {
    for (n = 0; n < length(product[0]); n++) {
      nprint(int_to_string(product[m][n]));
      nprint(" ");
    }
    nprint("\n");
  }
}

CONFIG {
  FS = "|";
}
```

The output of the demo is too verbose to put in the report, but can be produced by running

```
$ ./bawk.sh demo/demo.bawk demo/shuffled.txt
```