
Newbie

Programming for Dummies (Or highly inexperienced experts)

Who are We?



Clyde Bazile



John Anukem



Sebastien Siclait



Braxton Gunter



Terence Jacobs

Why Newbie?



```
1 #include "stdio.h"
2 #definee 3
3 #defineg (e/e)
4 #defineh ((g+e)/2)
5 #definef (e-g-h)
6 #definej (e*e-g)
7 #define k (j-h)
8 #definel(x) tab2[x]/h
9 #definem(n,a) ((n&(a))==a)
10
11 long tab1[]={ 989L,5L,26L,0L,88319L,123L,0L,9367L };
12 int tab2[]={ 4,6,10,14,22,26,34,38,46,58,62,74,82,86 };
13
14 main(m1,s) char *s; {
15     int a,b,c,d,o[k],n=(int)s;
16     if(m1==1){ char b[2*j+f-g]; main(l(h+e)+h+e,b); printf(b); }
17     else switch(m1--h){
18 case f:
19     a=(b=(c=(d=g)<<g)<<g)<<g;
20     return(m(n,a|c)|m(n,b)|m(n,a|d)|m(n,c|d));
21 case h:
22     for(a=f;a<j;++a)if(tab1[a]&&!(tab1[a]%((long)l(n))))return(a);
23 case g:
24     if(n<h)return(g);
25     if(n<j){n-=g;c='D';o[f]=h;o[g]=f;}
26     else{c='r'-'b';n-=j-g;o[f]=o[g]=g;}
27     if((b=n)>=e)for(b=g<<g;b<n;++b)o[b]=o[b-h]+o[b-g]+c;
28     return(o[b-g]*n+k-h);
29 default:
30     if(m1==e) main(m1-g+e+h,s+g); else *(s+g)=f;
```

```
#include <stdio.h>
int main()
{
    int c, first, last, middle, n, search, array[100];

    printf("Enter number of elements\n");
    scanf("%d",&n);

    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d",&array[c]);

    printf("Enter value to find\n");
    scanf("%d", &search);

    first = 0;
    last = n - 1;
    middle = (first+last)/2;

    while (first <= last) {
        if (array[middle] < search)
            first = middle + 1;
        else if (array[middle] == search) {
            printf("%d found at location %d.\n", search, middle+1);
            break;
        }
        else
            last = middle - 1;

        middle = (first + last)/2;
    }
    if (first > last)
        printf("Not found! %d is not present in the list.\n", search);

    return 0;
}
```



What can a Newbie do?

→ Comment

```
# I'm such a noob  
# enter code below
```

→ Operate

```
# Hi my name is newbie  
set x to 10  
set y to x  
set operators to "+, -, *, /, ^, %."  
print(operators)
```

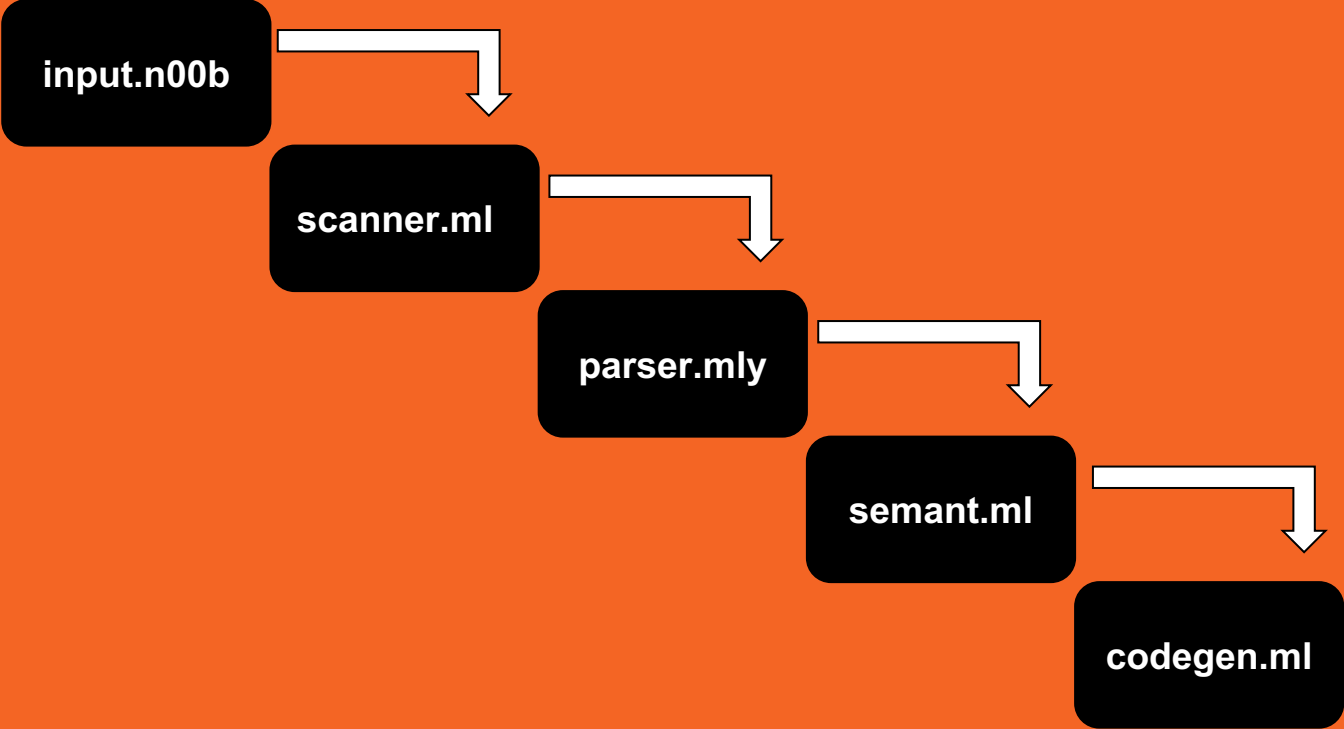
→ Infer

```
# Hi my name is newbie  
  
set x to 10 # type x: int  
set y to "hello" # type y: string
```

Syntax

```
1 define function plus_10 with params (a, b)
2   set c to 9 + 1
3   return a + b + c
4
5 define function main with no params
6   set x to 1
7   set y to 1
8   set out to plus_10(x, y)
```

Newbie Architecture



Language Features

- Easy syntax makes for simple implementation of algorithms
- Types:
 - Num
 - String
 - Bool
- Control Flow
 - If, Else, While, For
- Lists/Type Conversion/Coercion

- NoobTesting

```
testall.sh
79
80 Check() {
81     error=0
82     basename=`echo $1 | sed 's/.*\\.\\/'`
83     s/.n00b//''
84     reffile=`echo $1 | sed 's/.n00b$//''`
85     basedir=`echo $1 | sed 's/\\/[^\\]*$//''`
86
87     echo -n "$basename..."
88
89     echo 1>&2
90     echo "##### Testing $basename" 1>&2
91
92     generatedfiles=""
93
94     generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe ${basename}.out" &&
95     Run "$NEWBIE" $1 > "${basename}.out" &&
96     Compare ${basename}.out ${reffile}.out ${basename}.diff
97
98     # Report the status and clean up the generated files
99
100     if [ $error -eq 0 ] ; then
101     if [ $keep -eq 0 ] ; then
102         rm -f $generatedfiles
103     fi
104     echo "OK"
105     echo "##### SUCCESS" 1>&2
106     else
107     echo "##### FAILED" 1>&2
108     globalerror=$error
109     fi
110 }
111
112 while getopts kdph c; do
113     case $c in
114     k) # Keep intermediate files
115         keep=1
116         ;;
117     h) # Help
118         Usage
119         ;;
120     esac
121 done
```

```
test-precedence.n00b x
1  define function main with no params
2  set x to 2 + 3 * 4
3
4  print(x)
5
6  set x2 to (2 + 3) * 4
7
8  print(x2)
9
10 set y to 9 - 6 / 3
11
12 print(y)
13
14 set y2 to (9 - 6) / 3
15
16 print(y2)
17
```

```
test-precedence.out x
1  14
2  20
3  7
4  1
```


Lessons Learned

- Start early..... very early.
- Don't underestimate how long something will take, even if you think it'll be easy.
- Every line of OCaml counts..... EVERY LINE
- If you fail to test, your tests will fail.

Future Work

- Implementing more builtin classes with C.
- More builtin functions to handle walk through of staple algorithms

—

DEMO