Script started on Sat 18 Mar 2017 02:59:09 PM EDT

danwaba@itserver6:~/cs240/hw4$ cat main,.c[K[K[K.c

/\*

\* main.c

\*

\* reverse polish calculator

\*/

/\*

Daniel Werminghausen 3/19/17

cs240

Hw4

\*/

#include <stdio.h>

#include "calc.h" /\* header file \*/

#include <math.h>

#include <stdlib.h>

#define MAXOP 100

int main()

{

int type;

int op2;

char s[MAXOP];

while ((type = getop(s)) != EOF)

{

switch (type)

{

case NUMBER:

push(atoi(s));

break;

case '+':

push(pop() + pop());

break;

case '\*':

push(pop() \* pop());

break;

case '^':

push(pow(pop(), pop()));

break;

case '%':

push(pop() % pop());

break;

case '~':

push( ~ pop());

break;

case '-':

op2 = pop();

push(pop() - op2);

break;

case '/':

op2 = pop();

if (op2 != 0)

push(pop() / op2);

else

{

printf("error : zero divisor\n");

exit(1);

}

break;

case '\n':

printf("The answer is %i.\n", pop());

break;

default:

printf("error: unknown command %s\n", s);

exit(1);

}

}

}

danwaba@itserver6:~/cs240/hw4$ cat calc.h

#define NUMBER '0'

/\*

Daniel Werminghausen 3/19/17

cs240

Hw4

\*/

void push(int);

int pop(void);

int getop(char[]);

int getch(void);

void ungetch(int);

danwaba@itserver6:~/cs240/hw4$ cat stack.c

/\*

\* stack.c

\*

\* stack routines

\*/

/\*

Daniel Werminghausen 3/19/17

cs240

Hw4

\*/

#include <stdio.h>

#include "calc.h"

#define MAXVAL 100 /\* maximum depth of val stack \*/

/\* even better, we could make the following two defines "static"

and thus hide these data structures from the rest of the sources \*/

static int sp = 0; /\* next free stack position \*/

static int val[MAXVAL]; /\* value stack \*/

/\* push: push f onto value stack \*/

void push(int f)

{

if (sp < MAXVAL)

val[sp++] = f;

else

printf("error: stack full, can't push %d\n", f);

}

/\* pop: pop and return top value from stack \*/

int pop(void)

{

if (sp > 0)

return val[--sp];

else

{

printf("error: stack empty\n");

return 0;

}

}

danwaba@itserver6:~/cs240/hw4$ cat getch.c

/\*

\* getch.c

\*

\* getch returns one character from stdin

\* or from the buffer buf if it is not empty

\* ungetch puts one character in the buffer buf

\*/

/\*

Daniel Werminghausen 3/19/17

cs240

Hw4

\*/

#include <stdio.h>

#define BUFSIZE 100

char buf[BUFSIZE]; /\* buffer for ungetch \*/

int bufp = 0; /\* next free position in buf \*/

int getch(void) /\* get a (possibly pushed back) character \*/

{

return(bufp > 0) ? buf[--bufp] : getchar();

}

void ungetch(int c) /\* push character back on input \*/

{

if (bufp >= BUFSIZE)

printf("ungetch: too many character \n");

else

buf[bufp++] = c;

}

danwaba@itserver6:~/cs240/hw4$ cat getop.c

/\*

\* getop.c

\*

\* gets next token: operator or numeric operand

\*/

/\*

Daniel Werminghausen 3/19/17

cs240

Hw4

\*/

#include <stdio.h>

#include <ctype.h>

#include "calc.h"

int getop(char s[])

{

int i, c;

while ((s[0] = c = getch()) == ' ' || c == '\t')

;

s[1] = '\0';

if (!isdigit(c) && c != '.')

return c; /\* not a number \*/

/\* collect integer part in string s \*/

i = 0;

if (isdigit(c))

while (isdigit(s[++i] = c = getch()))

;

/\* collect fractional part in string s \*/

if (c == '.')

while (isdigit(s[++i] = c = getch()))

;

s[i] = '\0';

if (c != EOF)

ungetch(c);

return NUMBER;

}

danwaba@itserver6:~/cs240/hw4$ ls

calc.h getch.c getop.c hw4.doc main.o makefile2 stack.c typescript

calcit getch.o getop.o main.c makefile makefile.txt stack.o

danwaba@itserver6:~/cs240/hw4$ rm +[K\*.o

danwaba@itserver6:~/cs240/hw4$ ls

calc.h calcit getch.c getop.c hw4.doc main.c makefile makefile2 makefile.txt stack.c typescript

danwaba@itserver6:~/cs240/hw4$ ls -al[K[Kla

total 63

drwxr-sr-x 2 danwaba cs240-1G 512 Mar 18 15:00 .

drwxrws--- 6 danwaba cs240-1G 512 Mar 7 21:46 ..

-rw-r--r-- 1 danwaba cs240-1G 153 Mar 18 2017 calc.h

-rwxr-xr-x 1 danwaba cs240-1G 7876 Mar 18 14:56 calcit

-rw-r--r-- 1 danwaba cs240-1G 640 Mar 18 2017 getch.c

-rw-r--r-- 1 danwaba cs240-1G 677 Mar 18 2017 getop.c

-rw-r--r-- 1 danwaba cs240-1G 34816 Mar 6 14:24 hw4.doc

-rw-r--r-- 1 danwaba cs240-1G 935 Mar 18 2017 main.c

-rw-r--r-- 1 danwaba cs240-1G 1442 Mar 18 2017 makefile

-rw-r--r-- 1 danwaba cs240-1G 1458 Mar 18 2017 makefile2

-rw-r--r-- 1 danwaba cs240-1G 1326 Mar 18 14:28 makefile.txt

-rwxr-xr-x 1 danwaba cs240-1G 7876 Mar 18 14:11 .nfs000000000002203200000105

-rw-r--r-- 1 danwaba cs240-1G 0 Mar 18 14:55 .nfs000000000002203600000106

-rw-r--r-- 1 danwaba cs240-1G 738 Mar 18 2017 stack.c

-rw-r--r-- 1 danwaba cs240-1G 0 Mar 18 14:59 typescript

danwaba@itserver6:~/cs240/hw4$ make

make: Warning: File 'makefile' has modification time 7801 s in the future

gcc -m32 -c -o main.o main.c -lm

gcc -m32 -c -o getop.o getop.c -lm

gcc -m32 -c -o stack.o stack.c -lm

gcc -m32 -c -o getch.o getch.c -lm

gcc -m32 -g -o calcit main.o getop.o stack.o getch.o -lm

make: warning: Clock skew detected. Your build may be incomplete.

danwaba@itserver6:~/cs240/hw4$ ls

calc.h getch.c getop.c hw4.doc main.o makefile2 stack.c typescript

calcit getch.o getop.o main.c makefile makefile.txt stack.o

danwaba@itserver6:~/cs240/hw4$ ./calcit

16 292 % 292 16 / 291^H 2 ^ + ! ~

The answer is 2147483643.

q

error: unknown command q

danwaba@itserver6:~/cs240/hw4$ exit

exit

Script done on Sat 18 Mar 2017 03:01:09 PM EDT