rpcalc (version 1 : novice)

rpcalc.c

```
1: #include <stdio.h>
2: #include <stdlib.h>
3: #include <ctype.h>
4: #define MAXOP 100
5: #define NUMBER '0'
6: #define BUFSIZE 100
7: #define MAXVAL 100

8: int getop(char []);
9: void push(double);
10:double pop(void);
11:int getch(void);
12:void ungetch(int);
13:void stackError(char msg[]);

14:char buf[BUFSIZE];
15:int bufp = 0;
16:int sp = 0;
17:double val[MAXVAL];
18:int errnum = 0;

19:main()
20:{   /* reference
21:       EOF,MAXOP,NUMBER,errnum,stderr,getop(),push(),pop(),printf(),
22:       fprintf(),atof() */
23:}

24:int getop(char s[])
25:{
26:    /* reference EOF, NUMBER, getch(), ungetch(), isdigit() */
27:}

28:int getch(void)
29:{
30:    /* reference bufp, buf, getchar() */
31:}

32:void ungetch(int c)
33:{
34:    /* reference BUFSIZE, bufp, buf, stderr, fprintf() */
35:}

36:void push(double f)
37:{
38:    /* reference MAXVAL, sp, val, stackError() */
39:}

40:double pop(void)
41:{
42:    /* reference sp, val, stackError() */
43:}

44:void stackError(char msg[])
45:{
46:    /* reference errnum, stderr, fprintf() */
47:}
```

rpcalc (version 2 : neat and strict)

main.c

```
1:  #include <stdio.h>
2:  #include <stdlib.h>
3:  #define MAXOP 100
4:  #define NUMBER '0'

5:  int getop(char []);
6:  void push(double);
7:  double pop(void);

8:  int errnum = 0;

9:  main()
10: { /* reference
11:      EOF,MAXOP,NUMBER,errnum,stderr,getop(),push(),pop(),printf(),fprintf(),atof() */
12: }
```

getop.c

```
13: #include <stdio.h>
14: #include <ctype.h>
15: #define NUMBER '0'

16: int getch(void);
17: void ungetch(int);

18: extern int errnum;

19: int getop(char s[])
20: {
21:         /* reference EOF, NUMBER, getch(), ungetch(), isdigit() */
22: }
```

getch.c

```
23: #include <stdio.h>
24: #define BUFSIZE 100

25: static char buf[BUFSIZE];
26: static int bufp = 0;

27: int getch(void)
28: {
29:         /* reference bufp, buf, getchar() */
30: }
31: void ungetch(int c)
32: {
33:         /* reference BUFSIZE, bufp, buf, stderr, fprintf() */
34: }
```

stack.c

```
35: #include <stdio.h>
36: #define MAXVAL 100

37: static void stackError(char msg[]);
38: static int sp = 0;
39: static double val[MAXVAL];

40: void push(double f)
41: {
42:         /* reference MAXVAL, sp, val, stackError() */
43: }
44: double pop(void)
45: {
46:         /* reference sp, val, stackError() */
47: }
48: static void stackError(char msg[])
49: {
50:        extern int errnum;
51:        /* reference errnum, stderr, fprintf() */
52: }
```

rpcalc (version 3 : pragmatic)

rpcalc.h

```
1: #define NUMBER '0'

2: int getop(char []);
3: void push(double);
4: double pop(void);
5: int getch(void);
6: void ungetch(int);

7: extern int errnum;
```

main.c

```
8: #include <stdio.h>
9: #include <stdlib.h>
10:#include "rpcalc.h"
11:#define MAXOP 100

12:int errnum = 0;

13:main()
14:{ /* reference
15:     EOF,MAXOP,NUMBER,errnum,stderr,getop(),push(),pop(),printf(),fprintf(),atof() */
16:}
```

getop.c

```
17:#include <stdio.h>
18:#include <ctype.h>
19:#include "rpcalc.h"

20:int getop(char s[])
21:{
22:        /* reference EOF, NUMBER, getch(), ungetch(), isdigit() */
23:}
```

getch.c

```
24:#include <stdio.h>
25:#define BUFSIZE 100

26:static char buf[BUFSIZE];
27:static int bufp = 0;

28:int getch(void)
29:{
30:        /* reference bufp, buf, getchar() */
31:}
32:void ungetch(int c)
33:{
34:        /* reference BUFSIZE, bufp, buf, stderr, fprintf() */
35:}
```

stack.c

```
36:#include <stdio.h>
37:#include "rpcalc.h"
38:#define MAXVAL 100

39:static void stackError(char msg[]);
40:static int sp = 0;
41:static double val[MAXVAL];

42:void push(double)
43:{
44:        /* reference MAXVAL, sp, val, stackError() */
45:}
46:double pop(void f)
47:{
48:        /* reference sp, val, stackError() */
49:}
50:static void stackError(char msg[])
51:{
52:        /* reference errnum, stderr, fprintf() */
53:}
```

Makefile (version 1)

```
#
# rpcalc (a simple reverse-polish calculator)
#

rpcalc: main.o getop.o stack.o getch.o
        gcc -o rpcalc main.o getop.o stack.o getch.o

main.o: main.c rpcalc.h
        gcc -c main.c

getop.o: getop.c rpcalc.h
        gcc -c getop.c

stack.o: stack.c rpcalc.h
        gcc -c stack.c

getch.o: getch.c
        gcc -c getch.c
```

Makefile (version 2)

```
#
# rpcalc (a simple reverse-polish calculator)
#
TARGET = rpcalc
CC = gcc
CFLAGS =
LIBS =
OFILES = getch.o getop.o main.o stack.o

$(TARGET): $(OFILES)
        $(CC) -o $@ $(CFLAGS) $(OFILES) $(LIBS)

main.o: main.c rpcalc.h
        $(CC) -c $*.c

getop.o: getop.c rpcalc.h
        $(CC) -c $*.c

stack.o: stack.c rpcalc.h
        $(CC) -c $*.c

getch.o: getch.c
        $(CC) -c $*.c
```