```
1
2  #define _MAIN_C
3  #ifdef _MAIN_C
4
5  /*
6  *  ================================================================================
   ========
7  *       Filename:   why_calculator.c
8  *    Description:   a simple calculator
9  *        Version:   1.1.2
10 *        Created:   2017.10.26 16:04:15
11 *      Time Used:   10h
12 *  Last Modified:   2017.10.27 11:44
13 *    Last Change:   formated the main fun, making it more flexible
14 *         Author:   伍瀚缘(Tree Wu), why2000@hust.edu.cn
15 *        Company:   Huazhong University of Science and Technology
16 *  ================================================================================
   ========
17 */
18 //更新方向：
19 //1.增加负数的识别与读取（-a=(0-a)）          fixed
20 //2.对于多余运算符的识别                  fixed
21 //3.对于输入非特定字母命令的识别            fixed
22 //4.对于负数开奇整数次方的支持
23 #include<why_calculator.h>
24
25 //异常安全指示器
26 //0:正常结束，1:继续运行，2:跳过此次计算，其他:异常结束
27 int execstatus = 1;
28 //精度，表示保留的小数位数
29 int preci=15;
30
31 //真正的执行函数
32 void exectute(void) {
33     char input[MAXSIZE];
34     char repol[MAXSIZE];
35     char buf[MAXSIZE];
36     double result;
37     baseoutput();
38     while (execstatus) {//不为0时继续运行
39         if (execstatus != 2 && execstatus != 1) {//遇到异常错误时
40             printf("未知错误，请按任意键结束程序\n");
41             system("pause");
42             exit(-1);
43         }
44         execstatus = readinput(input,buf);//读取输入
45         if (execstatus == 2) {//遇到跳过本次计算请求时输出结束语并请求下一次输
             入
46             endingoutput();
47             execstatus = 1;
48             continue;
49         }
50         translate(input, repol);//转换为逆波兰表达式
51         if (execstatus == 2) {//请求下一次输入
```

```
52              endingoutput();
53              execstatus = 1;
54              continue;
55          }
56          //puts(repol);//检测逆波兰转换是否正确，非调试时注释
57
58          result = calculate(repol);
59          if (execstatus == 2) {//请求下一次输入
60              endingoutput();
61              execstatus = 1;
62              continue;
63          }
64
65          printf("%s = ", buf);
66          outputresult(result);
67          endingoutput();
68      }
69
70
71
72  }
73  #define NOW 1
74  #ifdef NOW//调试时停用
75  //表面主函数
76  int main(void) {
77      exectute();
78      return 0;
79  }
80
81  #endif //NOW
82
83  #endif //_MAIN_C
84
85
86
```