# 4 编译预处理实验

## 4.1 实验目的

（1）掌握文件包含、宏定义、条件编译和assert宏的使用；

（2）练习使用集成开发环境中的调试功能：单步执行、设置断点、观察变量值。

（3）熟悉多文件编译技术

## 4.2 实验内容

**4.2.1 程序改错**

下面是用宏来计算平方差、交换两数的源程序.在这个源程序中存在若干错误，要求对该程序进行调试修改，使之能够正确完成指定任务。

/\*实验4-1改错与跟踪调试题程序：计算平方差、将换两数\*/

1. #include<stdio.h>
2. #define SUM a+b
3. #define DIF a-b
4. #define SWAP(a,b) a=b,b=a
5. int main()
6. {
7. int a,b;
8. printf("Input two integers a, b:");
9. scanf("%d%d", &a,&b);
10. printf("\nSUM=%d\n the difference between square of a and square of b is:%d",SUM, SUM\*DIF);
11. SWAP(a,b);
12. printf("\nNow a=%d,b=%d\n",a,b);
13. return 0;
14. }

**解答：**

1. 语法错误修改：
2. 第2，3行的宏定义应该添加括号，正确形式为：

#define SUM (a+b)  // 加上括号

#define DIF (a-b)

2) 第4行的a=b,b=a，无法实现交换功能，正确形式为：

#define SWAP(a,b) a=SUM, b=DIF, a=DIF; // 更改为替换法

1. 错误修改后运行结果如图4-1所示。

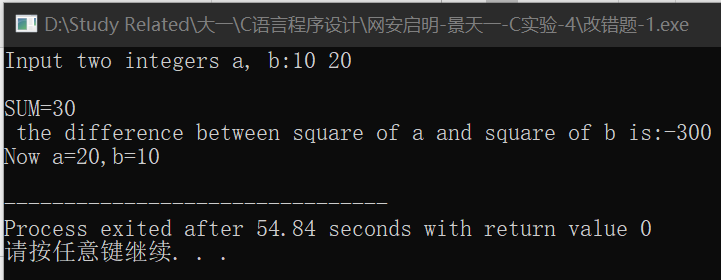


图4-1 实验1-1修改后运行结果截图

**4.2.2 程序修改替换**

下面是用函数实现求三个数中最大数、计算两浮点数之和的程序。在这个源程序中存在若干语法和逻辑错误。

要求：

对这个例子程序进行调试修改，使之能够正确完成指定任务；

1. 对这个例子程序进行调试修改，使之能够正确完成指定任务；
2. 用带参数的宏替换函数max，来实现求最大数的功能。

/\*实验4-2程序修改替换题程序\*/

1. #include<stdio.h>
2. int main(void)
3. {
4. int a, b, c;
5. float d, e;
6. printf("Input three integers:");
7. scanf("%d %d %d",&a,&b,&c);
8. printf("\nThe maximum of them is %d\n",max(a,b,c));
9. printf("Input two floating point numbers:");
10. scanf("%f %f",&d,&e);
11. printf("\nThe sum of them is %f\n",sum(d,e));
12. return 0;
13. }
14. int max(int x, int y, int z)
15. {
16. int m=z;
17. if (x>y)
18. if(x>z) m=x;
19. else
20. if(y>z) m=y;
21. return m;
22. }
23. float sum(float x, float y)
24. {
25. return x+y;
26. }

**解答：**

1. 对这个例子程序进行调试修改，使之能够正确完成指定任务；

在文件开始处声明需要定义的函数，正确形式如下：

int max(int x, int y, int z) ; // 未声明函数

1. 用带参数的宏替换函数max，来实现求最大数的功能。

实现形式如下：

#define MAX(x,y,z) ((x>y)?((x>z)?x:z):((y>z)?y:z)) // 输出最大值

1. 优化和替换后程序的源代码如下：
2. #include<stdio.h>
3. #define MAX(x,y,z) ((x>y)?((x>z)?x:z):((y>z)?y:z)) // 输出最大值
4. int max(int x, int y, int z) ; // 声明函数
5. float sum(float x, float y) ;
6. int main(void)
7. {
8. int a, b, c;
9. float d, e;
10. printf("Input three integers:");
11. scanf("%d %d %d",&a,&b,&c);
12. // printf("\nThe maximum of them is %d\n",max(a,b,c));
13. printf("\nThe maximum of them is %d\n",MAX(a,b,c));
14. printf("Input two floating point numbers:");
15. scanf("%f %f",&d,&e);
16. printf("\nThe sum of them is %f\n",sum(d,e));
17. return 0;
18. }
20. int max(int x, int y, int z)
21. {
22. int m=z;
23. if (x>y)
24. if(x>z) m=x;
25. else
26. if(y>z) m=y;
27. return m;
28. }
29. float sum(float x, float y)
30. {
31. return x+y;
32. }

优化和替换后运行结果如图4-2所示。

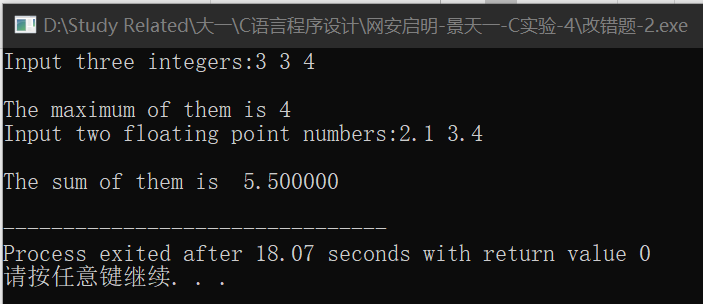


图4-2 实验4-2优化替换后运行结果截图

**4.2.3 跟踪调试**

下面程序利用R计算圆的面积s，以及面积s的整数部分。现要求：

（1）修改程序，使程序编译通过且能运行；

（2）单步执行。进入函数integerl\_fraction时，watch窗口中x为何值？在返回main时, watch窗口中i为何值？

（3）修改程序，使程序能输出面积s值的整数部分（要求四舍五入），不会输出错误信息assertion failed。

/\*实验4-3跟踪调试题程序利用R计算圆的面积s\*/

1. #define R
2. int main(void)
3. {
4. float r, s;
5. int s\_integer=0;
6. printf ("Input a number: ");
7. scanf("%f",&r);
8. #ifdef R
9. s=3.14159\*r\*r;
10. printf("Area of round is: %f\n",s);
11. s\_integer=integer\_fraction(s);
12. assert((s-s\_integer)<0.5);
13. printf("The integer fraction of area is %d\n", s\_integer);
14. #endif
15. return 0;
16. }
17. int integer\_fraction(float x)
18. {
19. int i=x;
20. return i;
21. }

**解答：**

1. 程序改错：
2. 在文件开始处添加头文件声明#include<stdio.h>
3. 在文件开始处补充函数声明；
4. 单步执行：

输入r为2，进入函数integerl\_fraction时，watch窗口中x为12.5663605，在返回main时, watch窗口中i为12。

1. 修改程序：

第十二行，源程序无法实现四舍五入，修改如下：

if (s-s\_integer>0.5) // 四舍五入

s\_integer++;

1. 源程序清单如下：
2. #include<stdio.h> // 改1：头文件
3. #define R
4. int integer\_fraction(float x); // 补充函数声明
5. int main(void)
6. {
7. float r, s;
8. int s\_integer=0;
9. printf ("Input a number: ");
10. scanf("%f",&r);
11. #ifdef R
12. s=3.14159\*r\*r;
13. printf("Area of round is: %f\n",s);
14. s\_integer=integer\_fraction(s);
15. // assert((s-s\_integer)<0.5); 这样无法实现四舍五入
16. if (s-s\_integer>0.5) // 四舍五入
17. s\_integer++;
18. printf("The integer fraction of area is %d\n", s\_integer);
19. #endif
20. return 0;
21. }
22. int integer\_fraction(float x)
23. {
24. int i=x;
25. return i;
26. }

运行结果如图4-3所示：

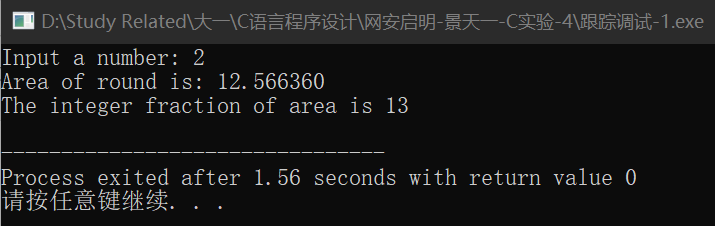


图4-3 跟踪调试题的运行结果图

符合预期结果，说明程序运行正确。

**4.2.4 程序设计**

1. 三角形的面积是，其中，a,b,c为三角形的三边，要求编写程序用带参数的宏来计算三角形的面积。定义两个带参数的宏，一个用来求s，另一个用来求area。

**解答：**

1. 解题思路：

利用math.h头文件中的pow()函数进行开方计算；为保证精确度，计算半周长s时应采用浮点数计算以保证精确度。

1. 程序清单：
2. #include<stdio.h>
3. #include<math.h>
4. #define S(a,b,c) (a+b+c)/2.0
5. #define AREA(a,b,c) pow((S(a,b,c)\*(S(a,b,c)-a)\*(S(a,b,c)-b)\*(S(a,b,c)-c)),0.5)
6. int main(void) {
7. int a=0,b=0,c=0 ;
8. printf("请输入三角形的三边：") ;
9. while(scanf("%d %d %d",&a,&b,&c)!=EOF) {
10. if ((a+b>c)&&(a+c>b)&&(b+c>a)) {
11. printf("Area is %f\n",AREA(a,b,c)) ;
12. }
13. else {
14. printf("%d,%d,%d三边无法组成三角形！\n",a,b,c);
15. }
16. printf("请输入三角形的三边：") ;
17. a=0;
18. b=0;
19. c=0;
20. }
21. return 0 ;
22. }

3）测试

（a） 测试数据：

程序设计题1的测试数据如表4-1所示。

表4-1 编程题1的测试数据

|  |  |  |
| --- | --- | --- |
| 测试用例 | 输入数据 | 预计输出 |
| 用例1 | 3，4，5 | 6.000000 |
| 用例2 | 2，2，2 | 2，2，2三边无法组成三角形 |
| 用例3 | ^Z | 结束 |

（b） 对应测试测试用例1，2，3的运行结果如图4-4所示。

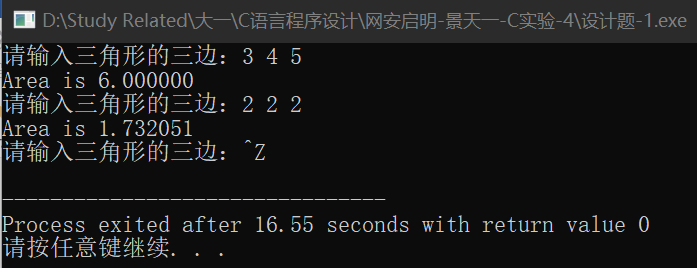


图4-4 程序设计题1的测试用例1，2，3的运行结果

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

1. 用条件编译方法来编写程序。输入一行英文字符序列，可以任选两种方式之一输出：一为原文输出；二为变换字母的大小写后输出。例如小写‘a’变成大写‘A’，大写‘D’变成小写‘d’，其他字符不变。用#define命令控制是否变换字母的大小写。例如，#define CHANGE 1 则输出变换后的文字，若#define CHANGE 0则原文输出。

**解答：**

1. 解题思路

解题思路如图4-5所示。

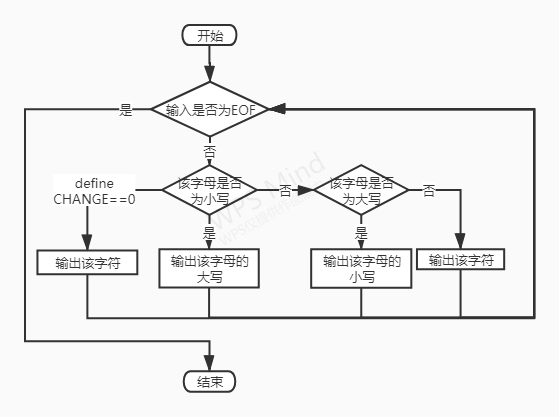


图4-5 程序设计题2算法流程图

1. 程序清单
2. #include<stdio.h>
3. #define CHANGE 0
4. char ch=0 ;
5. int main(void) {
6. while(scanf("%c",&ch)!=EOF) {
7. #if CHANGE==1
8. if (ch>=65 && ch<=90)
9. putchar(ch + 32) ;
10. else if (ch>=97 && ch<=122)
11. putchar(ch - 32) ;
12. else
13. putchar(ch) ;
14. #elif CHANGE==0
15. putchar(ch) ;
16. #endif
17. }
18. return 0 ;
19. }
20. 测试
21. 测试数据

程序设计题2测试数据如表4-2所示：

表4-2 编程题2的测试数据

|  |  |  |
| --- | --- | --- |
| 测试用例 | 输入数据 | 预计输出 |
| 用例1 | aSdFgH  (CHANGE=1) | AsDfGh |
| 用例2 | ASdFgH  (CHANGE=0) | aSdFgH |
| 用例3 | ^Z | 结束 |

1. 对应测试测试用例1的运行结果如图4-6所示。

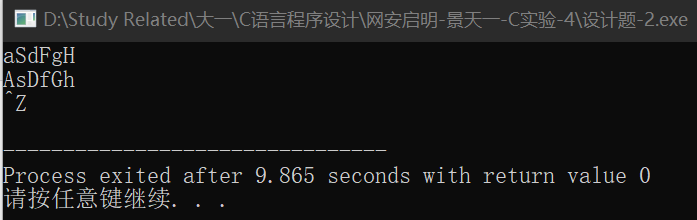


图4-6 程序设计题2的测试用例1的运行结果

1. 对应测试测试用例2, 3的运行结果如图4-7所示。

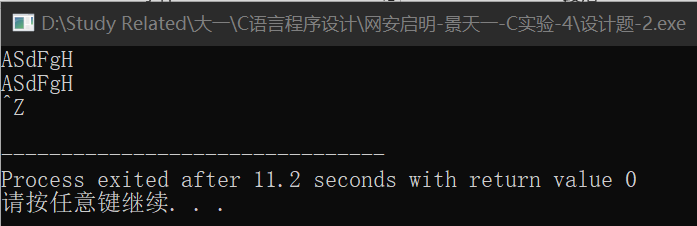


图4-7 程序设计题2的测试用例2, 3的运行结果

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

**4.2.5 选做题**

假设一个C程序由file1.c和file2.c两个源文件及一个file.h头文件组成，file1.c、file2.c和file.h的内容分别如下所述。试编辑该多文件C程序，补充file.h头文件内容，然后编译和链接。然后运行最后生成的可执行文件。

**解答：**

1. 解题思路：

在Dev C++中生成一个项目文件，通过头文件链接的方式链接外部参数和函数。

1. 程序清单
2. /\*头文件file.h的内容\*/
3. #include<stdio.h>
4. extern int x;
5. extern int y;
6. extern char ch;
7. void func1(void);
8. /\*源文件file1.c的内容\*/
9. #include "file.h"
10. int x,y;
11. char ch;
12. int main(void) {
13. x=10;
14. y=20;
15. ch=getchar();
16. printf("in file1 x=%d,y=%d,ch is %c\n",x,y,ch);
17. func1();
18. return 0;
19. }
20. /\*源文件file2.c的内容为：\*/
21. #include"file.h"
22. void func1(void) {
23. x++;
24. y++;
25. ch++;
26. printf("in file2 x=%d,y=%d,ch is %c\n",x,y,ch);
27. }
28. 测试

输入字符a，预计输出：

in file1 x=10,y=20,ch is a

in file2 x=11,y=21,ch is b

编译运行后file1.c运行结果如图4-8所示。

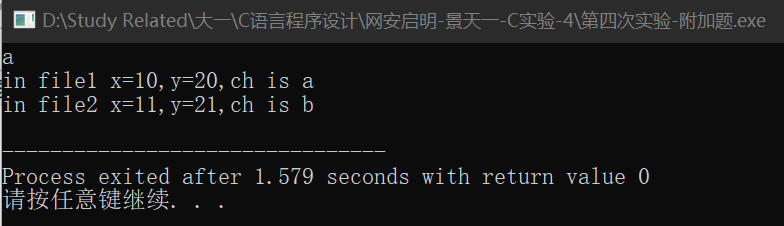


图4-8 选做题的运行结果

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

## 4.3 实验小结

本次实验主要练习了编译预处理的相关内容，包括头文件的声明，自定义文件的链接，宏定义，带参数的宏定义。头文件声明是C/C++程序的重要组成部分，许多库函数为程序编写提供了很大的方便；自定义文件链接为开发者自行编写库，对大型项目进行模块化设计和开发管理提供了可能；宏定义在一定程度上简化了重复出现参数，同时可以更加清楚的表示其含义。

在编写选做题的过程中，我在链接文件时遇到了困难，询问助教后通过使用搭建DevC++项目的形式来解决文件链接问题。但这也引出了我的疑问：如何在不适用该项目，不依赖IDE的情况下在多平台实现多文件链接？

参考文献

[1] 曹计昌,卢萍,李开. C语言程序设计, 北京：科学出版社, 2013

[2] 卢萍,李开,王多强等. C语言程序设计典型题解与实验指导, 北京：清华大学出版社, 2019