

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 软件工程2003班**

**学 号： U202010851**

**姓 名： 侯皓斐**

**指导教师： 唐赫**

**报告日期： 2020.11.28**

**软件工程**

**目 录**

[**4 编译预处理实验 1**](#_Toc404837920)

[4.1 实验目的 1](#_Toc404837921)

[4.2 实验内容 1](#_Toc404837922)

[4.3 实验小结 20](#_Toc404837923)

# 4 编译预处理试验

## 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 源程序改错**

**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。

1 /\*实验4-1改错与跟踪调试题程序：计算平方差、将换两数\*/

2 #include<stdio.h>

3 #define SUM a+b

4 #define DIF a-b

5 #define SWAP(a,b) a=b,b=a

6 int main()

7 {

8 int a,b;

9 printf("Input two integers a, b:");

10 scanf("%d%d", &a,&b);

11 printf("\nSUM=%d\n the difference between square of a and square of b is:%d",SUM, SUM\*DIF);

12 SWAP(a,b);

13 printf("\nNow a=%d,b=%d\n",a,b);

14 return 0;

15}

**解答：**

（1）错误修改：

1) 第3行为保证计算正确性，应使用括号，正确形式为：

#define SUM (a+b)

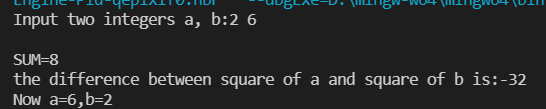
2) 第4行为保证计算正确性，应使用括号，正确形式为：

#define DIF (a-b)

3) 第5行swap不能完成交换的命令，使用交错相减的方法（如实验一），正确形式为：

#define SWAP(a,b) a=b-a,b=b-a,a=a+b

（2）错误修改后运行结果：



**4.2.2 源程序修改替换**

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。

/\*实验4-2程序修改替换题程序\*/

#include<stdio.h>

int main(void)

{

int a, b, c;

float d, e;

printf("Input three integers:");

scanf("%d %d %d",&a,&b,&c);

printf("\nThe maximum of them is %d\n",max(a,b,c));

printf("Input two floating point numbers:");

scanf("%f %f",&d,&e);

printf("\nThe sum of them is %f\n",sum(d,e));

return 0;

}

int max(int x, int y, int z)

{

int m=z;

if (x>y)

if(x>z) m=x;

else

if(y>z) m=y;

return m;

}

float sum(float x, float y)

{

return x+y;

}

要求：（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；

**解答：**

1) 第3行之前为保证通过编译，应对函数进行声明，正确形式为：

int max(int ,int ,int );//进行声明

float sum(float ,float );

2) 第22-26行为保证计算正确性，应正确分析if语句结构，增加花括号，正确形式为：

#define DIF (a-b)

3) 第5行swap不能完成交换的命令，使用交错相减的方法（如实验一），正确形式为：

int max(int x, int y, int z)

{

int m=z;

if (x>y) {

if(x>z) m=x;

}

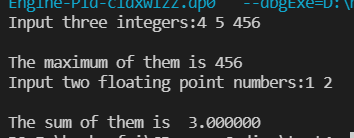
else {

if(y>z) m=y;

}

return m;

}



（2）用带参数的宏替换函数max，来实现求最大数的功能。

**解答：**

我们观察到计算三个数的最大值需要进行多次分类判断，源程序使用if语句，改为带参数的宏之后可以转换为三目运算来进行分类判断。

设需判断x,y,z三个数的最大值。

我们可以先比较x与y。

若x>y，则y不可能是最大值，选出x，z中的最大值即为最大值。

同理，若x<y，则x不可能是最大值，选出y，z中的最大值即为最大值。

宏可如下编写 #define max(a,b,c) (a>b? (a>c? a:c):(b>c? b:c))

因此可编写程序如下：

/\*实验4-2程序修改替换题程序\*/

#include<stdio.h>

#define max(a,b,c) (a>b? (a>c? a:c):(b>c? b:c))

float sum(float ,float );

int main(void)

{

int a, b, c;

float d, e;

printf("Input three integers:");

scanf("%d %d %d",&a,&b,&c);

printf("\nThe maximum of them is %d\n",max(a,b,c));

printf("Input two floating point numbers:");

scanf("%f %f",&d,&e);

printf("\nThe sum of them is %f\n", sum(d,e));

return 0;

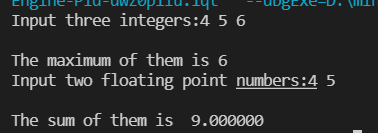
}

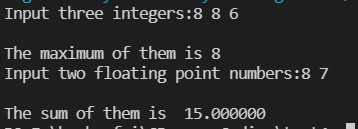
float sum(float x, float y)

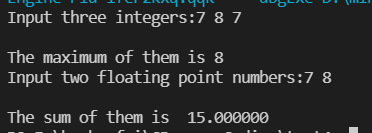
{

return x+y;

}







**4.2.3 程序跟踪调试**

下面程序利用R计算圆的面积s，以及面积s的整数部分。

/\*实验4-3跟踪调试题程序利用R计算圆的面积s\*/

#define R

int main(void)

{

float r, s;

int s\_integer=0;

printf ("Input a number: ");

scanf("%f",&r);

#ifdef R

s=3.14159\*r\*r;

printf("Area of round is: %f\n",s);

s\_integer=integer\_fraction(s);

assert((s-s\_integer)<0.5);

printf("The integer fraction of area is %d\n", s\_integer);

#endif

return 0;

}

int integer\_fraction(float x)

{

int i=x;

return i;

}

现要求：

1. 修改程序，使程序编译通过且能运行；

**错误修改：**

1) 第2行之前为保证通过编译，应对使用的库引入，正确形式为：

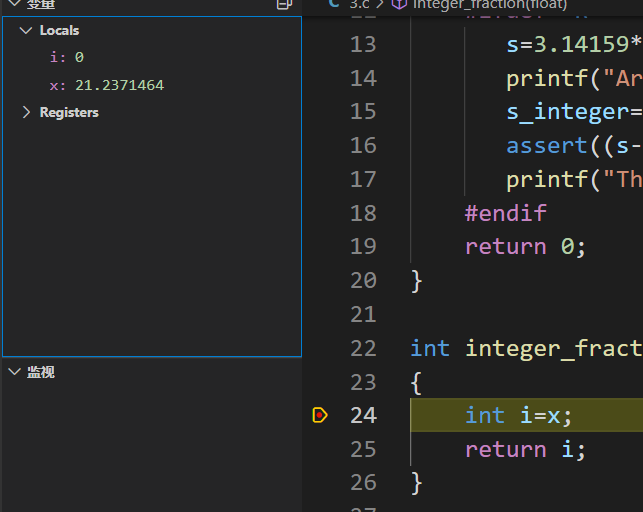
#include <stdio.h>//引用

2) 第4行之前为保证通过编译，应对需要使用的函数声明，正确形式为：

int integer\_fraction(float );//进行声明

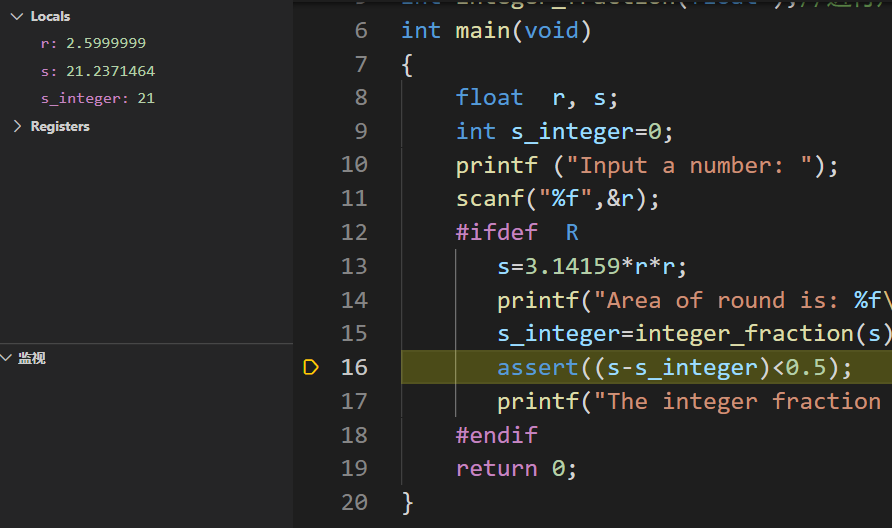
1. 单步执行。进入函数integerl\_fraction时，watch窗口中x为何值？在返回main时, watch窗口中i为何值？设输入5，观察刚执行完“scanf("%d",&k);”语句时，sum、k的值是多少？

**解答：**



当输入x = 2.6时，

进入函数integerl\_fraction时，watch窗口中x为21.2371461



返回main时, watch窗口中不能看到i的值，原因在于i为函数integerl\_fraction内的局部变量，执行结束后i的内存即被释放，不能被main函数使用，也不能被编译器发现。

1. 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。

**解答：**

已知断言函数arrset，用于在调试过程中捕捉程序的错误。assert() 会对表达式expression进行检测：如果expression的结果为 0（条件不成立），那么断言失败，表明程序出错，assert() 会向标准输出设备（一般是显示器）打印一条错误信息，并调用 abort() 函数终止程序的执行。

因此我们需要删除（1）题16行的assert((s-s\_integer)<0.5);语句。这样才不会输出错误信息assertion failed。

改为if判断语句，用以进行四舍五入。

若(s-s\_integer)>=0.5 s\_integer++;即可

完整代码如下：

/\*实验4-3跟踪调试题程序利用R计算圆的面积s\*/

#include <stdio.h>//引用

#include <assert.h>

#define R

int integer\_fraction(float );//进行声明

int main(void)

{

float r, s;

int s\_integer=0;

printf ("Input a number: ");

scanf("%f",&r);

#ifdef R

s=3.14159\*r\*r;

printf("Area of round is: %f\n",s);

s\_integer=integer\_fraction(s);

if((s-s\_integer)>=0.5)

s\_integer++;

printf("The integer fraction of area is %d\n", s\_integer);

#endif

return 0;

}

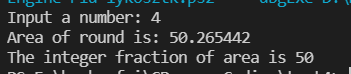
int integer\_fraction(float x) {

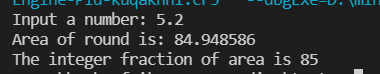
int i=x;

return i;

}

运行截图如下：





**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）本题为简单的顺序流程题目，关键点在于带参数宏的编写。

可构建算法流程图如下：

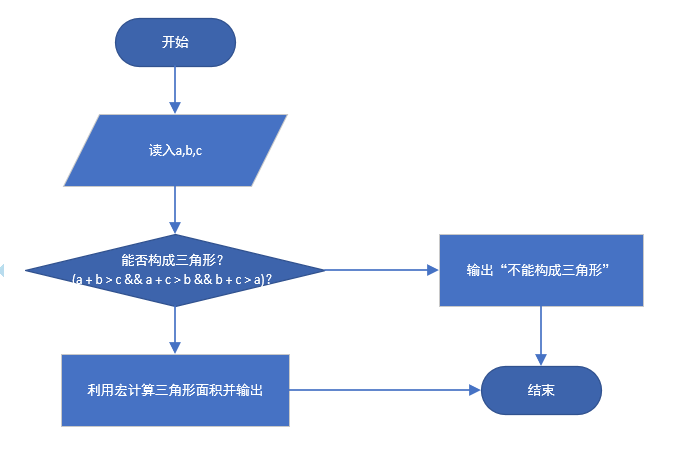


图4-1 编程题1的程序流程图

而编写宏计算三角形面积需要借助海伦－秦九韶公式

假设在平面内，有一个三角形，边长分别为a、b、c，三角形的面积S可由以下公式求得：。而公式里的p为半周长（周长的一半）：。编写带参数的宏p(a,b,c)和S(a,b,c)计算p和S。

对带参数的宏，在展开过程中不仅要进行字符串替换，还要用实参去替换形参。带参宏定义的一般形式为：#define 宏名(形参列表) 字符串

2）源程序清单

#include <stdio.h>

#include <math.h>

#define s(a,b,c) ((a+b+c) / 2.0)

#define area(a,b,c) sqrt(s(a,b,c) \* (s(a,b,c) - a) \* (s(a,b,c) - b) \* (s(a,b,c) - c))

int main () {

float a, b, c;

scanf("%f%f%f", &a, &b, &c);

if(a + b > c && a + c > b && b + c > a)

printf("S = %f\n", area(a, b, c));

else

printf("can not from a triangle\n");

return 0;

}

3）测试

（a） 测试数据：

3 4 5

5 12 13

6 6 6

100 1 1

（b） 对应测试数据的运行结果截图





图4-2 编程题1的测试运行结果

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。

（2）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：

本题需要利用条件编译方法，条件编译是指预处理器根据条件编译指令，有条件地选择源程序代码中的一部分代码作为输出，送给编译器进行编译。主要是为了有选择性地执行相应操作，防止宏替换内容（如文件等）的重复包含。

#if 如果条件为真，则执行相应操作

#elif 如果前面条件为假，而该条件为真，则执行相应操作

#else 如果前面条件均为假，则执行相应操作

#endif 结束相应的条件编译指令

#ifdef 如果该宏已定义，则执行相应操作

#ifndef 如果该宏没有定义，则执行相应操作

其中，#if-#else-#endif在本题中可以使用

#if CHANGE == 0

原文输出

#else

改换大小写后输出（可参考实验1编程题）

#endif

功能为：如果#if后的条件表达式为真，则程序段 1 被选中，否则程序段 2 被选中。

2）程序清单

#include <stdio.h>

#define CHANGE 1

int main () {

char str[10000];

gets(str);

#if CHANGE == 0

printf("%s", str);

#elif CHANGE == 1

for(int i = 0; str[i]; i++) {

if(str[i] >= 'a' && str[i] <= 'z')

str[i] = str[i] - 'a' + 'A';

else if(str[i] >= 'A' && str[i] <= 'Z')

str[i] = str[i] - 'A' + 'a';

}

printf("%s", str);

#endif

return 0;

}

3）测试

（a） 测试数据：

根据题目要求以及各种可能出现的错误情况构造数据如表4-1所示。

表4-1 编程题2的测试数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试  用例 | 编译 | 输 入 | 理 论 结 果 | 运 行 结 果 |
| CHANGE | 文本 |
| 例1 | 0 | Hello the World | Hello the World | Hello the World |
| 例2 | 1 | Hello the World | hELLO THE wORLD | hELLO THE wORLD |
| 例3 | 1 | Love the world! AND hello THE world123 | lOVE THE WORLD! and HELLO the WORLD123 | lOVE THE WORLD! and HELLO the WORLD123 |

（b） 对应测试测试用例1的运行结果如图4-3所示。



图4-3 编程题2的测试用例一的运行结果

对应测试测试用例2的运行结果如图4-4所示。



图4-4 编程题2的测试用例二的运行结果

对应测试测试用例3的运行结果如图4-5所示。



图4-5 编程题2的测试用例三的运行结果

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。

（3）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。

/\*源文件file1.c的内容\*/

#include "file.h"

int x,y; /\* 外部变量的定义性说明 \*/

char ch; /\* 外部变量的定义性说明 \*/

int main(void)

{

x=10;

y=20;

ch=getchar();

printf("in file1 x=%d,y=%d,ch is %c\n",x,y,ch);

func1();

return 0;

}

/\*源文件file2.c的内容为：\*/

#include "file.h"

void func1(void)

{

x++;

y++;

ch++;

printf("in file2 x=%d,y=%d,ch is %c\n",x,y,ch);

}

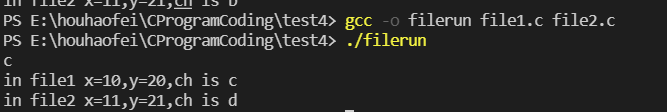
**解答：**

1. 解题思路：

头文件.h文件可以实现所需函数的声明，定义的外部变量可以跨文件，举例来说，如果文件a.c需要引用b.c中变量int v，就可以在a.c中声明extern int v，然后就可以引用变量v。能够被其他模块以extern修饰符引用到的变量通常是全局变量。

为保证编译通过且能够执行需要声明函数void func1(void)，使源文件file2.c实现。也需要声明外部变量int x,y;char ch;为了避免头文件被重复包含，应学会使用#ifndef语句，保证头文件编译时只被包含一次。

而我们要想编译和运行多文件的程序时，应使用命令行输入如下命令。



2）程序清单

/\*源文件file1.c的内容\*/

#include "file.h"

int x,y; /\* 外部变量的定义性说明 \*/

char ch; /\* 外部变量的定义性说明 \*/

int main(void)

{

x=10;

y=20;

ch=getchar();

printf("in file1 x=%d,y=%d,ch is %c\n",x,y,ch);

func1();

return 0;

}

/\*源文件file2.c的内容为：\*/

#include "file.h"

void func1(void)

{

x++;

y++;

ch++;

printf("in file2 x=%d,y=%d,ch is %c\n",x,y,ch);

}

/\*头文件file.h的内容为：\*/

#ifndef \_\_FILE\_H\_\_ //防止重复包含

#define \_\_FILE\_H\_\_

#include <stdio.h>

extern int x,y; /\* 外部变量的定义性说明 \*/

extern char ch; /\* 外部变量的定义性说明 \*/

void func1();

#endif

3）测试

（a） 测试数据：

根据题目要求以及各种可能出现的错误情况构造数据如表4-2所示。

表4-2 编程题3的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试  用例 | 输 入 | 理 论 结 果 | 运 行 结 果 |
| ch |
| 例1 | a | in file1 x=10,y=20,ch is a  in file2 x=11,y=21,ch is b | in file1 x=10,y=20,ch is a  in file2 x=11,y=21,ch is b |
| 例2 | D | in file1 x=10,y=20,ch is D  in file2 x=11,y=21,ch is E | in file1 x=10,y=20,ch is D  in file2 x=11,y=21,ch is E |

（b） 对应测试测试用例1的运行结果如图4-6所示。

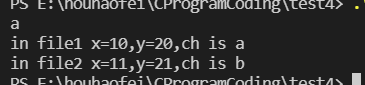


图4-6 编程题3的测试用例一的运行结果

对应测试测试用例2的运行结果如图4-7所示。

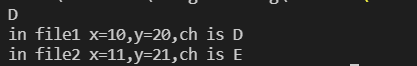


图4-7 编程题3的测试用例二的运行结果

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

主要叙述实验过程中遇到的问题，如何解决的，通过分析、结果问题后的体会。

本次实验通过改错题，程序修改替换题，编程题，结合上网搜索资料，较深入的掌握了宏定义，带参数的宏的编写。在跟踪调试题中再次强化了集成开发环境中的调试功能：单步执行、设置断点、观察变量值的能力，了解了第一次遇到的assert宏的使用。通过编程题2了解了条件编译的相关知识，对编译编译预处理的相关知识有了更深入的理解。编程题3卡了我很久，在与同学交流后，明白了自己的误区，外部变量也许在头文件中声明，且需要加extern修饰符，在同学的帮助下，使用命令行完成了多文件编译技术。