# 实验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、assert宏的使用；

（2）练习带参数的宏定义、条件编译的使用；

（3）练习assert宏的使用；

（4）使用Turbo C 2.0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。

## 4.2 实验内容

**4.2.1 源程序改错题**

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语法和逻辑错误。要求在计算机上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 #include "stdio.h"

2 #define SUM a+b

3 #define DIF a-b

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

5 void main

6 {

7 int b, t;

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 }

解答:

1. 错误修改:
2. 第2行的宏定义中a+b应该用括号括起来，否则预处理时可能产生未知结果，正确形式为：

#define SUM (a+b)

1. 第3行的宏定义中a-b应该用括号括起来，否则预处理时可能产生未知结果，正确形式为：

#define DIF (a-b)

1. 第4行的宏定义中逻辑存在问题，不能实现两数交换，应该更改，正确形式为：

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

1. 第5行声明主函数时应在void main后面加上括号（），正确形式为：

void main()

1. 第7行定义整形时没有定义a，多定义了t，正确形式为：

int b, a;

1. 第9行的scanf语句中%d与%d之间不能够加上逗号”,”，应去掉，正确形式为：

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

1. 第12行的printf拼写错误，不能将p写成大写形式，应使用p，正确形式为：

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

错误修改后源程序:

#include "stdio.h"

#define SUM (a+b)

#define DIF (a-b)

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

void main()

{

int b, a;

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

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

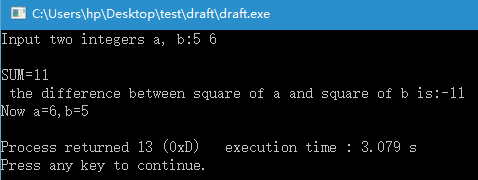
printf("\nSUM=%d\n the difference between square of a and square of b is:%d",SUM, SUM\*DIF);

SWAP(a,b);

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

}

1. 错误修改后运行结果:



**4.2.2 源程序修改替换题**

下面是用函数实现求三个数中最大数、计算两数之和的程序，在这个源程序中存在若干语法和逻辑错误。

要求：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。

1 void main(void)

2 {

3 int a, b, c;

4 float d, e;

5 printf("Enter three integers:");

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

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

8

9 printf("Enter two floating point numbers:");

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

解答：

1）调试修改程序以使其正常运行完成指定任务：

1. 错误修改：
2. 第1行声明主函数前未加上头文件，正确形式为：

#include<stdio.h>

1. 第1行声明主函数前未声明自定义函数，正确形式为：

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

float sum(float x, float y);

1. 第6、10行的scanf语句中%d与%d之间不能够加上逗号”,”，应去掉，正确形式为：

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

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

错误修改后源程序:

#include<stdio.h>

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

float sum(float x, float y);

void main(void)

{

int a, b, c;

float d, e;

printf("Enter three integers:");

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

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

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

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

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

}

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

{

int t;

if (x>y)

t=x;

else

t=y;

if (t<z)

t=z;

return t;

}

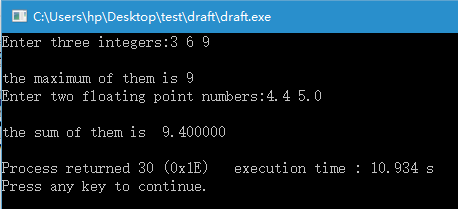
float sum(float x, float y)

{

return (x+y);

}

1. 错误修改后运行结果：



2）用带参数的宏替换函数max，来实现求最大数：

将max函数替换为#define max(a,b,c) a>=b?(a>=c)?a:c:(b>=c)?b:c。

修改后的源程序：

#include<stdio.h>

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

float sum(float x,float y);

void main(void)

{

int a, b, c;

float d, e;

printf("Enter three integers:");

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

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

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

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

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

}

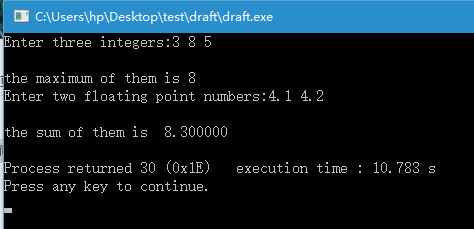
float sum(float x, float y)

{

return (x+y);

}

修改后的运行结果：



**4.2.3 跟踪调试题**

下面程序利用R计算圆的面积s，以及面积s的整数部分。

1 #define R

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

13 assert((s-s\_integer)<1.0);

14 #endif

15 }

16

17 int integer\_fraction(float x)

18 {

19 int i=x;

20 return i;

21 }

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decimal\_fraction时watch窗口中x为何值？在返回main时, watch窗口中i为何值？

3）排除错误，使程序能正确输出面积s值的整数部分，不会输出错误信息assertion failed。

解答：

1）修改程序，使其编译通过且能运行：

1. 错误修改：
2. 第1行宏定义前未加上头文件，正确形式为：

#include<stdio.h>

#include<assert.h>

1. 第2行声明主函数前未声明自定义函数，正确形式为：

int integer\_fraction(float x);

错误修改后源程序:

#include<stdio.h>

#include<assert.h>

#define R

int integer\_fraction(float);

void 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);

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

assert((s-s\_integer)<1.0);

#endif

}

int integer\_fraction(float x)

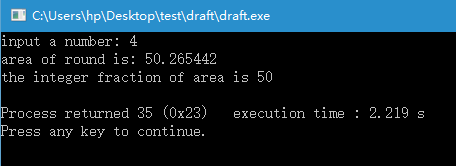
{

int i=x;

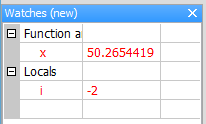
return i;

}

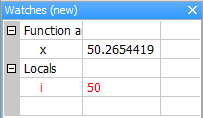
1. 修改后程序运行结果：



2）单步执行。进入函数decimal\_fraction时watch窗口中x的值如图：



在返回main时, watch窗口中i的值如图：



3）问题1）中修改后的程序已经能正确输出面积s值的整数部分，且不会输出错误信息assertion failed。

**4.2.4 编程设计题**

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

解答：

1. 解题思路：
2. 宏定义S(a,b,c)与AREA(a,b,c)分别计算三角形半周长S与面积AREA，即#define S(a,b,c) (a+b+c)/2

#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))

1. 输入a、b、c，判断a、b、c是否符合要求：如果a>0且b>0且c>0，转2.1，如果遇文件尾，转4，否则转3；
   1. 使用宏定义完成三角形半周长S与面积AREA的计算并输出，即 printf("%.0lf %lf\n",S(a,b,c),AREA(a,b,c));
   2. 转2，等待输入新信息。
2. 显示输入错误信息并转2，等待输入新信息；
3. 结束。
4. 程序清单：

#include<stdio.h>

#include<math.h>

#define S(a,b,c) (a+b+c)/2

#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(void)

{

double a,b,c;

while(scanf("%lf%lf%lf",&a,&b,&c)!=EOF)

{

if(a>0&&b>0&&c>0)

printf("%.0lf %lf\n",S(a,b,c),AREA(a,b,c));

else

printf("Wrong Input!\n");

}

}

1. 测试
2. 测试数据：

为方便分析测试结果，选择a、b、c分别为3、4、5，选取一组造成输出错误提示的数据-5、5、5，再选取一组较为一般的数据35、36、46。如表1-1所示。

表4-1 编程题4.2.4.(1)的测试数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试  用例 | 程 序 输 入 | | | 理 论 结 果 |
| a | b | c |
| 用例1 | 3 | 4 | 5 | 计算结果 半周长6、面积6.000000 |
| 用例2 | -5 | 5 | 5 | 提示输入错误信息 |
| 用例3 | 35 | 36 | 46 | 计算结果 半周长59、面积621.810612 |

1. 对应测试测试用例1、2、3的运行结果如图4-1所示。

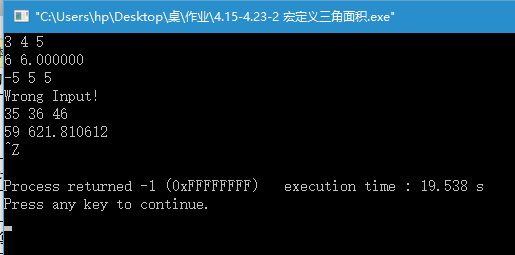


图4-1 编程题4.2.4.(1)的测试用例一的运行结果

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。

（2）用条件编译方法来编写程序。输入一行电报文字，可以任选两种输出：一为原文输出；二为变换字母的大小写（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：
2. 提示并输入电报个数N，一行电报文字为一个电报；
3. 输入电报文字并用宏定义CHANGE(c)对电报第一个字符进行判断：若CHANGE(c[0])=1，将电文内的字母大小写转换；否则不对电文进行处理；
4. 输出处理后的电文，转2，直至处理完N个电文；
5. 结束。
6. 程序清单：

#include<stdio.h>

#define CHANGE(c) c%2

int main()

{

int i,N;

char c[500];

scanf("%d",&N);

getchar();

while(N--)

{

fgets(c,500,stdin);

if(CHANGE(c[0]))

{

for(i=0;c[i]!='\0';i++)

{

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

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

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

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

}

}

printf("%s",c);

}

return 0;

}

1. 测试
2. 测试数据：

为方便分析测试结果，选取输入电文分别为hello! World# !、abcd efg ihk p!、Uq;V42? / \Ziggbee。如表4-2所示。

表4-2 编程题4.2.4.(2)的测试数据

|  |  |  |
| --- | --- | --- |
| 测试  用例 | 程 序 输 入 | 理 论 结 果 |
| 3组电文输入 |
| 用例1 | hello! World# ! | 输出结果hello! World# ! |
| 用例2 | abcd efg ihk p! | 输出结果ABCD EFG IHK P! |
| 用例3 | Uq;V42? / \Ziggbee | 输出结果uQ;v42? / \zIGGBEE |

1. 对应测试测试用例1、2、3的运行结果如图4-2所示。

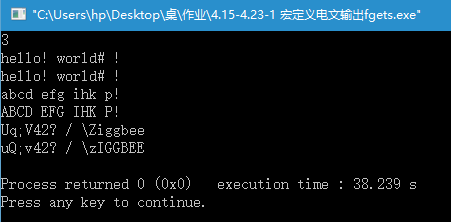


图4-2 编程题4.2.4.(2)的测试用例1、2、3的运行结果

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。

## 4.3 自设题

**（1）**自设实验题目：用宏定义的方法判断输入年份是否为闰年

**（2）**实验目的：通过设计实验程序，理解预编译原理并熟悉宏定义的使用。

**（3）**实验程序：

#include<stdio.h>

#define JUDGE(x) ((x%4==0)&&(x%100!=0)||(x%400==0))

int main(void)

{

int x;

printf("Please input year:");

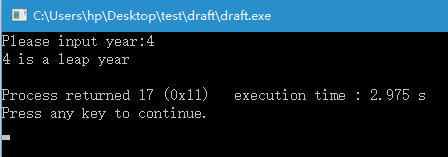
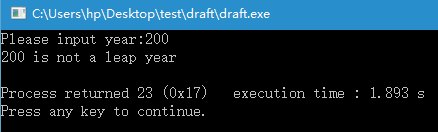
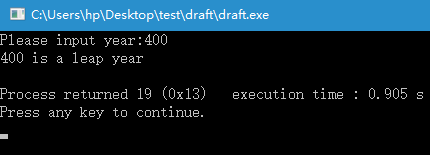
scanf("%d",&x);

if JUDGE(x) printf("%d is a leap year\n",x);

else printf ("%d is not a leap year\n",x);

}

**（4）**实验用例：如图

**（5）**实验结论：对三个实验用例，程序输出结果与预期结果一致，验证了程序的正确性。

## 4.4 实验小结

在实验中主要遇到的问题有：

1. 在4.2.1源程序改错中有关宏定义使用的错误较多，其中使用宏定义时未加括号是造成问题出现的常见原因之一且较难发现，会使程序无法达到预期目的；
2. 在1.2.2源程序修改替换时，出现了多种语句语法或者拼写上的错误，如printf中的字母p使用了大写，又如scanf中”%d %d %d”之间使用了逗号，都会使程序不能正常编译或运行错误；
3. 在解决问题4.2.2程序设计（2）使用带参宏定义替换函数判断最大值时，实现方式与函数相比有较大差异，需要熟练思路后才能更快解决问题。

体会：

1. 在编写程序时需严加注意格式方面的问题，并尽可能减少程序运行过程中可能出现的问题。如对函数进行引用时应当保证其拼写的准确，使用scanf时双引号内的内容间不应使用逗号，使用带参宏定义#define时尽量在后面声明的用于替换的语句两旁加上括号()，以避免程序逻辑出现错误。类似这样的错误有些并不会被编译器指出，从而可能造成调试困难。建议编写函数使程序更易调试，或使用单步调试以更准确地处理错误；
2. 善于使用宏定义、带参宏定义与条件编译能够在某些情况下提高代码的可移植性、稳定性、普适性，从而减少代码量。在熟练、准确使用的前提下，更要善于利用其解决实际遇到的问题或简化代码；
3. 在这次实验中又接触到了断言assert，熟练对他的使用能够使程序员对代码的调试更为方便准确，但若一个程序中出现过多的assert也容易造成代码的臃肿，这时可在头文件#include<assert.h>之前使用#define NDEBUG来禁用对assert的调用。