# 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

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
11. is:%d",SUM, SUM\*DIF);
12. SWAP(a,b);
13. Printf("\nNow a=%d,b=%d\n",a,b);
14. }

**解答：**

（1）错误修改：

1) 第1行的"stdio.h"，正确形式为：

<stdio.h>

2) 第4行的a=b,b=a，正确形式为：

t=a;a=b;b=t

3) 第5行的void main，正确形式为：

void main()

4)第6和7行添加a定义，正确形式为：

int a;

5) 第9行%d,%d中间无逗号，正确形式为：

%d%d

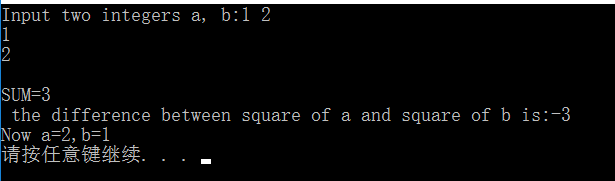
6) 第11行SUM DIF需加括号，正确形式为：

(SUM)\*(DIF)

7) 第13行Printf，正确形式为：

printf;

（2）错误修改后运行结果：



4.2.2 源程序修改替换

（1）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。

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;

}

**解答：**

1. 修改：

#include<stdio.h>

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

float sum(float x, float y);

float f;

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;

}

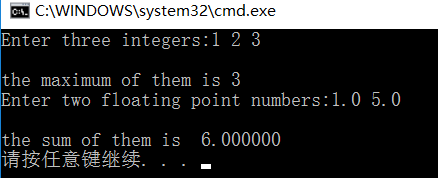
float sum(float x, float y)

{

return x+y;

}

运行结果：



**2）解答：**

#include<stdio.h>

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

float sum(float x, float y);

float f;

void main(void)

{

int a, b, c;

float d, e;

int m;

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));

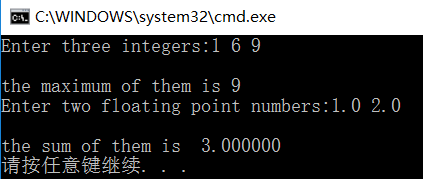
}

float sum(float x, float y)

{

return x+y;

}



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。

#define R

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。

1）

#include<stdio.h>

#include<assert.h>

#define R

int integer\_fraction(float x);

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);

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

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

#endif

}

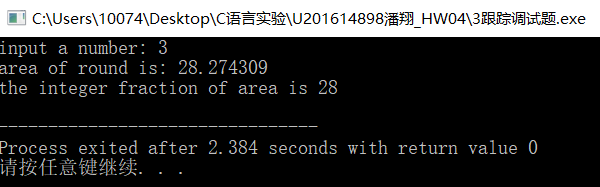
int integer\_fraction(float x)

{

int i = x;

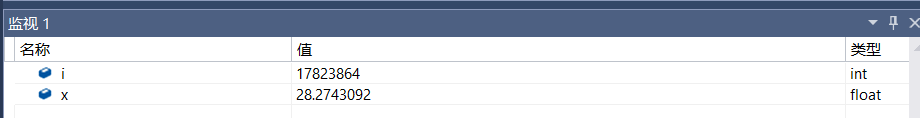
return i;

}



2）

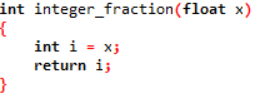
**进入**：



**出来：**



3）

进行转换 满足小于1.0条件

4.2.4 程序设计

（1）求三角形的面积是C:\Users\10074\AppData\Local\Temp\ksohtml\wpsAC24.tmp.png，其中C:\Users\10074\AppData\Local\Temp\ksohtml\wpsAC25.tmp.png，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） 算法流程如图4.1所示。



图4-1 编程题1的程序流程图

2）源程序清单

#include<stdio.h>

#include<math.h>

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

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

int main()

{

double a, b, c;

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

{

int s = s(a, b, c);

double area = area(a, b, c);

printf("%d ", s);

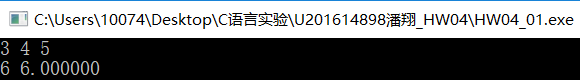
printf("%lf\n", area);

}

return 0;

}

3）测试



（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） 算法流程如图4.2所示。



图4-2 编程题2的程序流程图

2）源程序清单

#include<stdio.h>

#include<string.h>

#define CHANGE c%2

void Conversion();

char s[100];

int len;

int count = 0;

int i;

int main()

{

int N;

scanf("%d", &N);

getchar();

for (int i = 1; i <= N; i++)

{

fgets(s, 100, stdin);

len = strlen(s)-1;

//printf("%d\n", len);

char c = s[0];

if (CHANGE != 0)

{

Conversion();

}

else

{

printf("%s", s);

}

}

}

void Conversion()

{

for (int i = 0; i < len; i++)

{

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

{

s[i] = s[i] - 32;

}

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

{

s[i] = s[i] + 32;

}

}

printf("%s", s);

}

3）测试

（a） 测试数据：

hello !

world !

hello world!

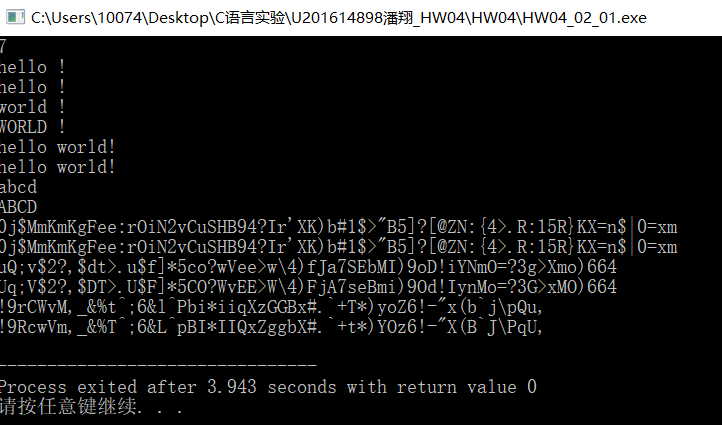
abcd

0j$MmKmKgFee:rOiN2vCuSHB94?Ir'XK)b#1$>"B5]?[@ZN:{4>.R:15R}KX=n$|0=xm

uQ;v$2?,$dt>.u$f]\*5co?wVee>w\4)fJa7SEbMI)9oD!iYNmO=?3g>Xmo)664

!9rCWvM,\_&%t^;6&l^Pbi\*iiqXzGGBx#.`+T\*)yoZ6!-"x(b`j\pQu,

（b） 对应测试数据的运行结果截图



## 4.3 自设题

**（1）**自设实验题目：

描述

T. Chur教大学的各类学生。每个U-student都有一个独特的学生识别号码（SIN）。 SIN s是范围为0≤s≤MaxSIN，MaxSIN = 106-1的整数。 T. Chur发现这个SIN的范围太大，无法识别她的团体。 对于每个组，她想要找到最小的正整数m，使得在组内，所有SIN减小模m是唯一的。

输入

在输入的第一行是单个正整数N，告诉测试用例（组）的数量。 每个案例从包含整数G（1≤G≤300）的一行开始：组中的学生人数。 以下G行各自包含一个SIN。 一组内的SIN是不同的，但不一定是排序的。

输出

对于每个测试用例，输出包含最小模数m的一行，使得模m减少的所有SIN都不同。

样例输入

2

1

124866

3

124866

111111

987651

样例输出

1

8

**（2）**实验目的：结合算法设计，熟悉宏定义memset实现方法

#define MEM(a,x) memset(a,x,sizeof(a))

**（3）**实验程序：

#include <iostream>

#include <cstdio>

#include <cstring>

#include <cstdlib>

#define MEM(a,x) memset(a,x,sizeof(a))

using namespace std;

const int MAXN = 1000010;

bool visit[MAXN];

bool remainder[MAXN];

int main()

{

int t;

int n;

int num[310];

scanf("%d",&t);

while(t--)

{

scanf("%d",&n);

memset(visit, 0, sizeof(visit));

int maxNum = 0;

int i;

for(i = 0; i < n; i++)

{

scanf("%d",&num[i]);

}

int j;

for(i = 0; i < n; i++)

{

for(j = 0; j < n; j++)

{

visit[abs(num[i] - num[j])] = true;

}

}

for(i = 1; ; i++)

{

if(!visit[i])

{

// cout<<"i= "<<i<<endl;

MEM(remainder,0);

bool flag = true;

for(j = 0; j < n; j++)

{

if(remainder[num[j] % i] == true)

{

flag = false;

break;

}

remainder[num[j] % i] = true;

}

if(flag == true)

{

printf("%d\n",i);

break;

}

}

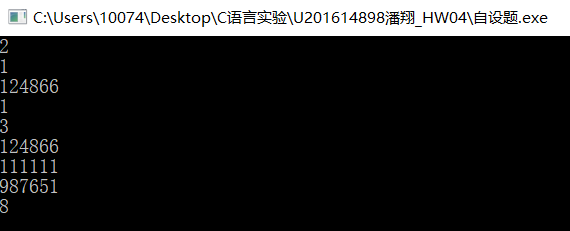
}

}

return 0;

}

**（4）**实验用例**：**



**（5）**实验结论：

对于多层FOR循环 和常用的内存初始化函数宏定义的熟悉 使结构清晰

#define FOR(i,f\_start,f\_end) for(int i=f\_start;i<=f\_end;++i)

#define MEM(x,i) memset(x,i,sizeof(x))

## 4.4 实验小结

**（1）**恰当使用宏定义使常用的常量不输入出错和结构清晰

**（2）**学习使用测试用例和有效的测试方法。

**（3）**注重细节，培养习惯才能在工程代码中少出错。

**（4）**结合算法和OJ题目，提高程序编写能力。