

**课 程 实 验 报 告**

**课程名称： C语言程序设计**

**专业班级： 物联网1601**

**学 号： U201614887**

**姓 名： 林子涵**

**指导教师： 刘芳**

**报告日期： 2017.6.8**

**计算机科学与技术学院**

目 录

[**1 表达式和标准输入与输出实验** 3](#_Toc484704885)

[1.1 实验目的 3](#_Toc484704886)

[1.2 实验内容 3](#_Toc484704887)

[1.3 实验小结 9](#_Toc484704888)

[**2 流程控制实验** 9](#_Toc484704889)

[2.1 实验目的 9](#_Toc484704890)

[2.2 实验内容 10](#_Toc484704891)

[2.3 实验小结 21](#_Toc484704892)

[**3 函数与程序结构实验** 21](#_Toc484704893)

[3.1 实验目的 21](#_Toc484704894)

[3.2 实验内容 22](#_Toc484704895)

[3.3 实验小结 29](#_Toc484704896)

[**4 编译预处理实验** 29](#_Toc484704897)

[4.1 实验目的 29](#_Toc484704898)

[4.2 实验内容 29](#_Toc484704899)

[4.3 实验小结 37](#_Toc484704900)

[**5 数组实验** 37](#_Toc484704901)

[5.1 实验目的 37](#_Toc484704902)

[5.2 实验内容 38](#_Toc484704903)

[5.3 实验小结 52](#_Toc484704908)

[**6 指针实验** 52](#_Toc484704909)

[6.1 实验目的 52](#_Toc484704910)

[6.2 实验内容 52](#_Toc484704911)

[6.3 实验小结 65](#_Toc484704912)

[**7 结构与联合实验** 66](#_Toc484704913)

[7.1 实验目的 66](#_Toc484704914)

[7.2 实验内容 66](#_Toc484704915)

[7.3 实验小结 80](#_Toc484704916)

[**8 文件实验** 80](#_Toc484704917)

[8.1 实验目的 80](#_Toc484704918)

[8.2 实验内容 80](#_Toc484704919)

[8.3 实验小结 85](#_Toc484704920)

[**9. 参考文献** 85](#_Toc484704921)

# 1 表达式和标准输入与输出实验

## 1.1 实验目的

（1）熟练掌握各种运算符的运算功能，操作数的类型，运算结果的类型及运算过程中的类型转换，重点是C语言特有的运算符，例如位运算符，问号运算符，逗号运算符等；熟记运算符的优先级和结合性；

（2）掌握getchar, putchar, scanf 和printf 函数的用法。

（3）掌握简单C程序（顺序结构程序）的编写方法。

## 1.2 实验内容

**1.2.1源程序改错题**

下面给出了一个简单C语言程序例程，用来完成以下工作：

1．输入华氏温度f，将它转换成摄氏温度c后输出；

2．输入圆的半径值ｒ，计算并输出圆的面积ｓ；

3．输入短整数ｋ、ｐ，将ｋ的高字节作为结果的低字节，ｐ的高字节作为结果的高字节，拼成一个新的整数，然后输出；

在这个例子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 PI 3.14159;

3 void main( void )

4 {

5 int f ;

6 short p, k ;

7 double c , r , s ;

8

9 /\* for task 1 \*/

10 printf(“Input Fahrenheit:” ) ;

11 scanf(“%d”, f ) ;

12 c = 5/9\*(f-32) ;

13 printf( “ \n %d (F) = %.2f (C)\n\n ”, f, c ) ;

14

15 /\* for task 2 \*/

16 printf("input the radius r:");

17 scanf("%f", &r);

18 s = PI \* r \* r;

19 printf("\nThe acreage is %.2f\n\n",&s);

20 /\* for task 3 \*/

21 printf("input hex int k, p :");

22 scanf("%x %x", &k, &p );

23 newint = (p&0xff00)|(k&0xff00)<<8;

24 printf("new int = %x\n\n",newint);

25

26 }

27

**解答：**

（1）错误修改：

1) 第2行的符号常量定义后不能有分号，正确形式为：

**#define PI 3.14159**

2) 第11行的scanf函数参数类型错误，正确形式为：

**scanf(“%d”,&f ) ;**

3) 第12行的c赋值表达式需强制类型转换，正确形式为：

**c = (double)5/9\*(f-32) ;**

4) 第17行的scanf函数占位符与参数类型不符，正确形式为：

**scanf("%lf", &r);**

5) 第19行的printf函数参数类型错误，正确形式为：

**printf("\nThe acreage is %.2f\n\n",s);**

6) 第22行的scanf函数占位符错误，正确形式为：

**scanf("%hx %hx", &k, &p );**

7) 第23行的newint未定义，正确形式为：

**short newint;**

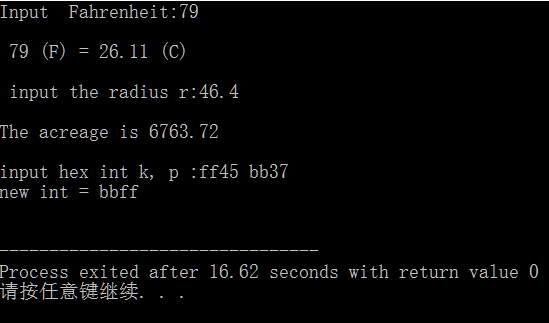
8)第23行的语句实现逻辑错误，正确形式为：

**newint = (p&0xff00)|(k&0xff00)>>8;**

9) 第24行的printf函数占位符错误，正确形式为：

**printf("new int = %hx\n\n",newint);**

（2）错误修改后运行结果：

****

**1.2.2、源程序修改替换题**

下面的程序利用常用的中间变量法实现两数交换，请改用不用第三个变量的交换法实现。

#include<stdio.h>

void main( )

{

int a, b, t;

printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

t=a, a=b, b=t;

prinf(“\na=%d,b=%d”,a,b);

}

**解答：**

利用位运算中x^x=0以及x^0=x这两个性质和异或运算的结合律，可用位运算实现两数交换，替换后的程序如下所示：

#include<stdio.h>

void main( )

{

int a, b;

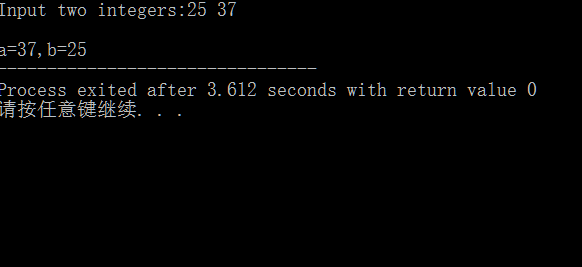
printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

a=a^b ;b=b^a;a=a^b;

printf(“\na=%d,b=%d”,a,b);

}

****

**1.2.3、编程设计题**

上机调试运行以下程序：

1. 编写一个程序，输入无符号短整数x，ｍ，ｎ（0 ≤ｍ≤ 15, 1 ≤ ｎ≤ 16-ｍ）,取出x从第ｍ位开始向左的ｎ位（ｍ从右至左编号为0～15），并使其向左端（第15位）靠齐。

**1)解题思路**:

1.输入x，m，n，

2.首先x>>m，将要处理的n位移动到最右；

3.再将上一步的结果左移16-n位，即：x<<=(16-n);

4.输出结果.

**2)源代码**

**#include<stdio.h>**

**int cases;**

**unsigned short x;**

**int m,n;**

**int main(){**

**scanf("%hu%d%d",&x,&m,&n);**

**x>>=m;**

**x<<=(16-n);**

**printf("%hu\n",x);**

**return 0;**

**}**

表1-1 编程题3的测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试  用例 | 程 序 输 入 | | | 理 论 结 果 | 运 行 结 果 |
| X | m | N |
| 用例1 | 65530 | 7 | 4 | 61440 | 截图 |
| 用例2 | 64720 | 13 | 1 | 32768 |  |
| 用例3 | 55791 | 13 | 2 | 32768 |  |

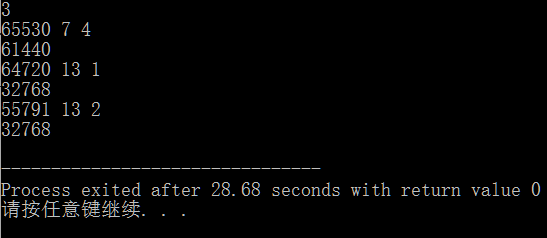


图1-1 编程题1的测试用例一的运行结果

1. IP地址通常是4个用句点分隔的小整数，如32.55.1.102。这些地址在机器中用无符号长整形表示。编写一个程序，以机器存储的形式读入一个32位的互联网IP地址，对其译码，然后用常见的句点分隔的4部分的形式输出。

**1)解题思路：**

1.首先输入无符号整型n，

2.利用n与逻辑尺的&运算依次取出n的4个字节；

3.再将上一步的四个结果分别右移对应位数使其右对齐;

4.按顺序输出结果并以点相间隔.

**2）源代码**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**unsigned int n;**

**int main(){**

**scanf("%u",&n);**

**int A=0x000000ff;**

**printf("%d.",(A&n));**

**A=0x0000ff00;**

**printf("%d.",(A&n)>>8);**

**A=0x00ff0000;**

**printf("%d.",(A&n)>>16);**

**A=0xff000000;**

**printf("%d\n",(A&n)>>24);**

**return 0;**

**}**

表1-2 编程题3的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试  用例 | 程 序 输 入 | 理 论 结 果 | 运 行 结 果 |
| n |
| 用例1 | 134744072 | 8.8.8.8 | 截图 |
| 用例2 | 676879571 | 211.92.88.40 |
| 用例3 | 16885952 | 192.168.1.1 |
| 用例4 | 16843009 | 1.1.1.1 |
| 用例5 | 26975347 | 115.156.155.1 |

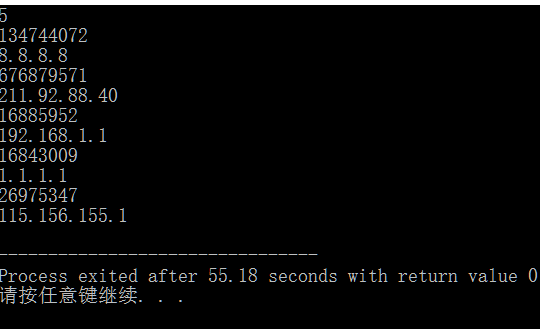


图1-2 编程题2的测试用例一的运行结果

## 1.3 实验小结

本次试验中遇到的问题有：改错题中short类型以16进制输出的占位符不知道，编程题中第三题我不知道是无符号整型高位先输出还是低位先输出。对于第一个问题我查阅了c语言课本，了解了各种类型及格式的占位符，对于输出格式的多样性有了深入的了解。对于第二个问题，我通过查找资料结合样例输入输出得出结论是低位先输出，通过这个题我对于IP地址的存储有了更加深入的理解，了解到压缩存储的意义。

# 2 流程控制实验

## 2.1、实验目的

（1）掌握复合语句、if语句、switch语句的使用，熟练掌握for、while、do-while三种基本的循环控制语句的使用，掌握重复循环技术，了解转移语句与标号语句。

（2）练习循环结构for、while、do-while语句的使用。

（3）练习转移语句和标号语句的使用。

（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。

## 2.2、实验内容

**2.2.1源程序改错题**

下面是计算s=n!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。例如，8！=40320。

1 #include <stdio.h>

2 void main(void)

3 {

4 int i,n,s=1;

5 printf("Please enter n:");

6 scanf("%d",n);

7 for(i=1,i<=n,i++)

8 s=s\*i;

9 printf("%d! = %d",n,s);

10}

11 return

**解答：**

（1）错误修改：

1) 第6行的scanf()参数错误，正确形式为：**scanf("%d",&n);**

2) 第7行的for()语句格式错误，正确形式为：**for(i=1;i<=n;i++)**

3) 第11行的return语句应在main函数内。

**2.2.2源程序修改替换题**

（1）修改第1题，分别用while和do-while语句替换for语句。

**解答：**

替换后的程序如下所示：

while 语句：

#include <stdio.h>

int main()

{

int i,n,s=1;

printf("Please enter n:");

scanf("%d",&n);

i=1;

while(i<=n){

s=s\*i;

i++;

}

printf("%d! = %d",n,s);

return 0;

}

do-while语句：

#include <stdio.h>

int main()

{

int i,n,s=1;

printf("Please enter n:");

scanf("%d",&n);

i=1;

do{

s=s\*i;

i++;

}while(i<=n);

printf("%d! = %d",n,s);

return 0;

}

（2）修改第1题，输入改为“整数S”，输出改为“满足n！≥S的最小整数n”。例如输入整数40310，输出结果为n=8。

**解答：**

#include <stdio.h>

int main()

{

int i,n,s;

printf("Please enter n:");

scanf("%d",&s);

for(n=1,i=1;n<s;i++,n\*=i);

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

return 0;

}

**2.2.3编程设计题**

（1）打印如下杨辉三角形。

1 /\*第0行 \*/

1 1 /\*第1行 \*/

1 2 1 /\*第2行 \*/

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

每个数据值可以由组合计算（表示第i行第j列位置的值），而的计算如下：

 (i=0,1,2,…)

 (j=0,1,2,3,…,i)

本程序中为了打印出金字塔效果，要注意空格的数目。一位数之间是3个空格，两位数之间有2个空格，3位数之间只有一个空格，程序编制过程中要注意区分。

1. **解题思路**:

1.利用杨辉三角的递推关系预处理出每位的值

2.输入n

3.循环输出空格及每位的值，两个数之间的空格数由后者确定

3.1 第i行行首空格数为2\*(n-i)

3.2 每行中维护两个变量分别代表要输出的数和前一个数

3.3每行最后一个数输出后换行.

3. 结束

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int n,space,first,second;**

**int yang[15][15];//存储杨辉三角的值**

**void init(){**

**int i,j;**

**yang[1][1]=1;**

**for(i=2;i<=12;i++)**

**for(j=1;j<=i;j++)**

**yang[i][j]=yang[i-1][j]+yang[i-1][j-1];**

**}**

**int calc(int x){**

**if(x>99) return 1;**

**if(x>9) return 2;**

**return 3;**

**}**

**int main(){**

**init();//初始化**

**int i,j,k;**

**while(scanf("%d",&n),n){**

**first=0;second=0;**

**space=2\*(n-1);//每行开始的空格数**

**for(i=1;i<=n;i++){**

**for(j=1;j<=i;j++){**

**if(j==1){**

**for(k=1;k<=space;k++)**

**putchar(' ');//输出每行开始的空格数**

**if(i==1) printf("%d\n",yang[i][j]);**

**else printf("%d",yang[i][j]);**

**first=yang[i][j];**

**}**

**else if(j==i){**

**second=yang[i][j];**

**for(k=1;k<=calc(second);k++)**

**putchar(' ');//以两数的后者为标准输出空格数**

**printf("%d\n",yang[i][j]);**

**}**

**else {**

**second=yang[i][j];**

**for(k=1;k<=calc(second);k++)**

**putchar(' ');//以两数的后者为标准输出空格数**

**printf("%d",yang[i][j]);**

**first=yang[i][j];**

**}**

**}**

**space-=2;//每行开始空格比上一行少2个**

**}**

**printf("\n");**

**}**

**return 0;**

}

3）**测试**

（a） 测试数据：n=5，n=2，n=10，n=11，n=0

（b） 对应测试运行结果如图所示。

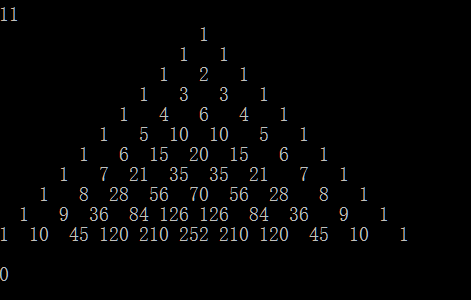


图2-1 编程题1的测试用例一的运行结果

（2）编写一个程序，将用户输入的任意正整数逆转，例如，输入1234，输出4321。

1) **解题思路**：

1.以字符串形式存储读入的整数

2.利用strlen()判断字符串长度，并从最后一位倒序输出

3. 结束

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**char s[100];**

**int main(){**

**while(scanf("%s",s)){**

**if(s[0]=='0') break;**

**//char \*ans=strrev(s);**

**for(int i=strlen(s)-1;i>=0;i--)**

**printf("%c",s[i]);**

**printf("\n");**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：n=123，n=1，n=123456，n=987654321，n=0

（b） 对应测试运行结果如图所示。

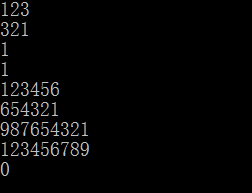


图2-2 编程题2的测试用例一的运行结果

（3）假设工资税金按以下方法计算：x<1000元，不收取税金;1000<=x<2000,收取5%的税金；2000<=x<3000,收取10%的税金；3000<=x<4000,收取15%的税金；4000<=x<5000,收取20%的税金；x>5000,收取25%的税金。编写一个程序，输入工资金额，输出应收取税金额度，要求分别用if语句和switch语句来实现。

1) **解题思路**：

1. 利用if或switch语句对x进行逐级判断，累加每一级别的税金，最后输出。

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**const double eps=1e-6;**

**double n,ans;**

**double min(double x,double y){**

**if(x-y>eps) return y;**

**return x;**

**}**

**int main(){**

**while(scanf("%lf",&n)){**

**ans=0.0;**

**if(abs(n)<eps) break;**

**if(n-5000>eps)**

**ans+=(n-5000.0)\*0.25;**

**if(n-4000>eps)**

**ans+=(min(n,5000.0)-4000.0)\*0.2;**

**if(n-3000>eps)**

**ans+=(min(n,4000.0)-3000.0)\*0.15;**

**if(n-2000>eps)**

**ans+=(min(n,3000.0)-2000.0)\*0.1;**

**if(n-1000>eps)**

**ans+=(min(n,2000.0)-1000.0)\*0.05;**

**printf("%.6lf\n",ans);**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：n=500，1000，1500……4500.

（b） 对应测试运行结果如图所示。

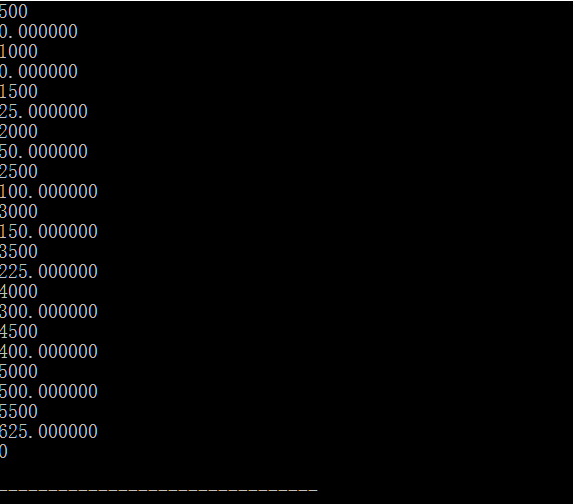


图2-3 编程题3的测试用例一的运行结果

4）编写一个程序，将输入的一行字符复制到输出，复制过程中将一个以上的空格字符用一个空格代替。

1) **解题思路：**

1. 循环读入每行。

2.每个字符用scanf(“%c”,&c)读入。

3.判断每个字符是否为空格，若是的话判断空格个数，用计数器保存已读到的空格数；若不是的话计数器清零。

2）**源代码**

/**/#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int n,cnt;**

**char c;**

**int main(){**

**scanf("%d",&n);**

**getchar();**

**for(int i=1;i<=n;i++){**

**while(scanf("%c",&c)){**

**if(c=='\n') break;**

**if(c==' '){**

**cnt++;**

**if(cnt==1) printf(" ");**

**}**

**else {**

**cnt=0;**

**printf("%c",c);**

**}**

**}**

**printf("\n");**

**cnt=0;**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：n=3

Hello world !

Hello world!

Romance is the glamour which turns the dust of everyday life into a golden haze.

（b） 对应测试运行结果如图所示。

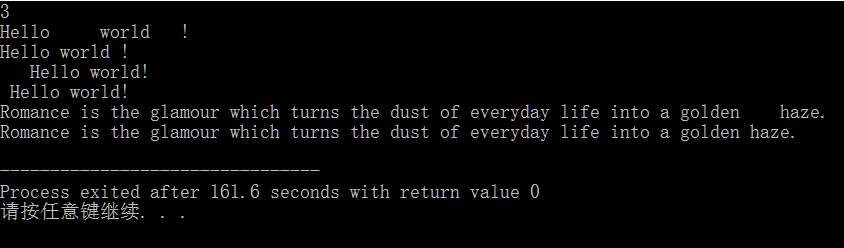


图2-4 编程题4的测试用例一的运行结果

**2.2.4选做题**

编写一个程序，用牛顿迭代法求方程f(x)=3x3-4x2-5x+13=0

1) **解题思路**：

1.在循环中利用牛顿迭代法公式反复迭代，循环退出条件为两次迭代结果满足精度要求。

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**const double eps=1e-6;**

**double f(double x){**

**return 3.0\*x\*x\*x-4.0\*x\*x-5.0\*x+13.0;**

**}**

**double f\_pie(double x){**

**return 9.0\*x\*x-8.0\*x-5.0;**

**}**

**double abss(double x){**

**if(x>eps) return x;**

**return -x;**

**}**

**double ans=0.0;**

**double ans1=0.0;**

**int main(){**

**while(1){**

**ans1=ans-f(ans)/f\_pie(ans);**

**if(abss(ans1-ans)<=eps) break;**

**ans=ans1;**

**}**

**printf("%.6lf\n",ans);**

**return 0;**

**}**

3）**测试**

（a） 测试数据：无

（b） 对应测试运行结果如图所示。

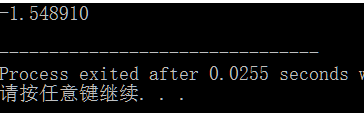


图2-5 选做题的测试用例一的运行结果

## 2.3 实验小结

本实验中杨辉三角题较复杂，我在完成过程中对于空格数的判断出现了些许的问题，最后我通过对样例的研读和找规律发现了空格数的规律。整数翻转题我采用了将整数认为是字符串的思路从而避免了对整数每一位的操作，简化了编程思路。通过这次实验我学到了如何使用循环、判断等流程控制语句，对C语言有了更好的理解

# 3 函数与程序结构实验

## 3.1实验目的

（1）熟悉和掌握函数的定义、声明；函数调用与参数传递方法；以及函数返回值类型的定义和返回值使用。

（2）熟悉和掌握不同存储类型变量的使用。

（3）熟悉多文件编译技术。

## 3.2、实验内容

**3.2.1 源程序改错**

下面是计算s=1!+2!+3!+…+n!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 void main(void)

3 {

4 int k;

5 for(k=1;k<6;k++)

6 printf("k=%d\tthe sum is %ld\n",k,sum\_fac(k));

7 }

8 long sum\_fac(int n)

9 {

10 long s=0;

11 int i;

12 long fac;

13 for(i=1;i<=n;i++)

14 fac\*=i;

15 s+=fac;

16 return s;

17 }

**解答：**

（1）错误修改：

1) 第2行缺少函数声明，正确形式为：**long sum\_fac(int n);**

2) 第12行的fac没有初始值，正确形式为：**long fac=1;**

3) 第13-15行缺少大括号

**3.2.2．源程序修改替换题**

（1）修改第1题中sum\_fac函数，使其计算量最小。

**解答：**

**#include<stdio.h>**

**long sum\_fac(int n);**

**int main()**

**{**

**int k;**

**for(k=1;k<6;k++)**

**printf("k=%d\tthe sum is %ld\n",k,sum\_fac(k));**

**return 0;**

**}**

**long s=0,fac=1;**

**long sum\_fac(int n)**

**{**

**fac\*=n;**

**return s+=fac;**

**}**

（2）修改第1题中sum\_fac函数，计算。

**解答：**

**#include<stdio.h>**

**double sum\_fac(int n);**

**int main()**

**{**

**int k;**

**for(k=1;k<6;k++)**

**printf("k=%d\tthe sum is %lf\n",k,sum\_fac(k));**

**return 0;**

**}**

**double sum\_fac(int n)**

**{**

**double s=0;**

**int i;**

**double fac=1;**

**for(i=1;i<=n;i++)**

**{ fac\*=i;**

**s+=(1/fac);}**

**return s;**

**}**

**3.2.3．跟踪调试题**

计算fabonacci数列前n项和的程序如下：

其中，long sum=0,\*p=&sum;声明p为长整型指针并用&sum取出sum的地址对p初始化。\*p表示引用p所指的变量（\*p即sum）。

#include<stdio.h>

long fabonacci(int);

void main(void)

{

int i,k;

long sum=0,\*p=&sum;

scanf("%d",&k);

for(i=1;i<=k;i++){

sum+=fabonacci(i);

printf("i=%d\tthe sum is %ld\n",i,\*p);

}

}

long fabonacci(int n)

{

if(n==1 || n==2)

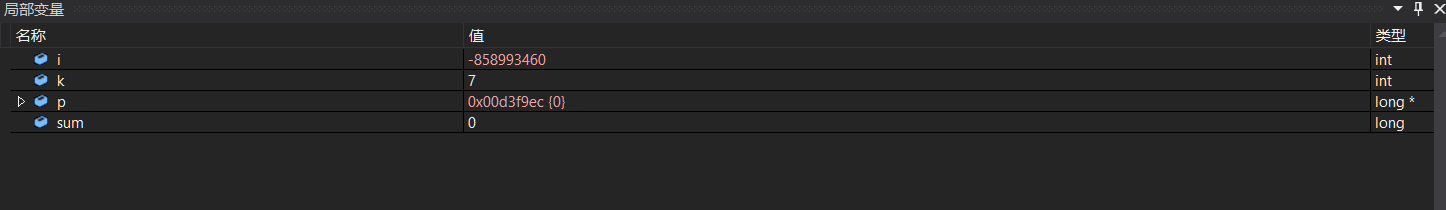
return 1;

else

return fabonacci(n-1)+fabonacci(n-2);

}

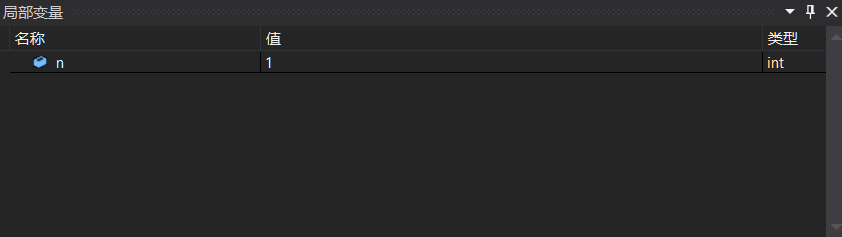
单步执行程序，观察p,i,sum,n值。

（1）刚执行完scanf("%d",&k);语句，p,i值是多少？

（2）从fabonacci函数返回后光条停留在哪个语句上？

**sum+=fabonacci(i);**语句上

（3）进入fabonacci函数，watch窗口显示的是什么？



（4）当i=3，从调用fabonacci函数到返回，n值如何变化？

n=3 ->n=2 ->n=1 ->n=3

**3.2.4．编程设计题**

（1）编程让用户输入两个整数，计算两个数的最大公约数并且输出之（要求用递归函数实现求最大公约数）。同时以单步方式执行该程序，观察递归过程。

1. **解题思路**:

1.利用欧几里得算法递归求解最大公约数

2.输入a，b

3.进入递归函数

3.1 若b==0，返回a

3.2 否则进行递归参数变为（b，a%b）

4. 输出结果

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int x,y;**

**int gcd(int a,int b){**

**if(b==0) return a;**

**return gcd(b,a%b);**

**}**

**int main(){**

**while(scanf("%d%d",&x,&y)){**

**if(x==0) break;**

**printf("%d\n",gcd(x,y));**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：a=10，b=20；a=40，b=70；a=36，b=24

（b） 对应测试运行结果如图所示。

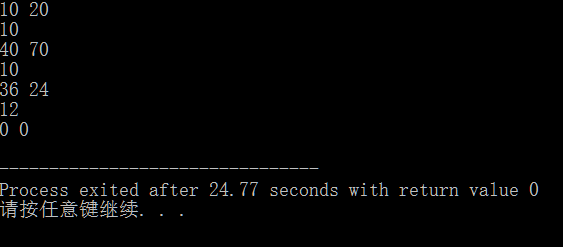


图3-1 编程题1的测试用例一的运行结果

（2）编程验证歌德巴赫猜想：一个大于等于4的偶数都是两个素数之和。

编写一个程序证明对于在符号常量BEGIN和END之间的偶数这一猜测成立。例如，如果BEGIN为10，END为20，程序的输出应为：

GOLDBACH'S CONJECTURE:

Every even number n>=4 is the sum of two primes.

10=3+7

12=5+7

……

20=3+17

1）**解题思路**:

1.利用线性筛法先将100内的素数筛出

2.输入begin和end，进入循环

3.对于循环的每个数从2开始进入第二层循环

3.1 循环变量为j，若j和i-j均为素数即输出

4. 结束

2）**源代码**

其地址目录为：

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int begin,end;**

**int prime[101];**

**void is\_prime(){**

**for(int i=2;i<=100;i++)**

**if(!prime[i]){**

**for(int j=2;j\*i<=100;j++)**

**prime[j\*i]=1;**

**}**

**}**

**int main(){**

**is\_prime();**

**while(scanf("%d%d",&begin,&end)){**

**if(begin==0) break;**

**printf("GOLDBACH'S CONJECTURE:\nEvery even number n>=4 is the sum of two primes\n");**

**for(int i=begin;i<=end;i++)**

**if(i%2==0)**

**for(int j=2;j<=i;j++)**

**if(!prime[j]&&(!prime[i-j])){**

**printf("%d=%d+%d\n",i,j,i-j);**

**break;**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：BEGIN=24，END=35

（b） 对应测试运行结果如图所示。

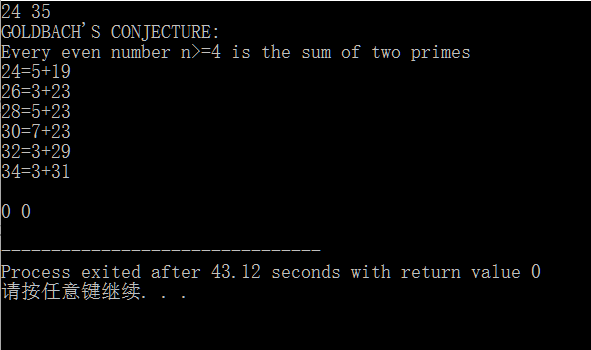


图3-2 编程题2的测试用例一的运行结果

## 3.3 实验小结

通过这次实验我掌握了一些基本的流程语句，，加强了一些基本语法的认识。

# 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 实验内容及要求

**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(void)

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：

1) 第2行宏定义加括号，正确形式为：**#define SUM (a+b)**

2) 第3行宏定义加括号，正确形式为：**#define DIF (a-b)**

3) 第7行缺少变量a的定义，正确形式为：**int b, t,a;**

4) 第12行Printf()函数错误，正确形式为：**printf()**

**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。

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：**

#include<stdio.h>

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

float sum(float x, float y);

#define max(x,y,z) x>y?(x>z? x:z):(y>z? y:z)

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;

}

**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；

**解答：**

**#include<stdio.h>**

**#include<assert.h>**

**#define R**

**int integer\_fraction(float x);**

**int main()**

**{**

**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)**

**{**

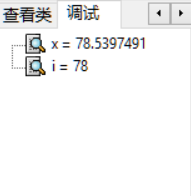
**int i=x;**

**return i;**

**}**

1. 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？

**解答：当输入5时，调试信息如下：**

****

1. 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。 第一步中已完成

**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。

1. **解题思路**:

1.利用带参数的宏定义s(a,b,c)

2.宏定义area(a,b,c,s)

3.调用宏输出结果

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**/\*定义计算面积的带参数的宏\*/**

**#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))**

**double a,b,c;**

**int x,y,z;**

**int main(){**

**while(scanf("%d%d%d",&x,&y,&z)!=EOF){**

**a=x;b=y;c=z;**

**if((x+y+z)%2) printf("%.1lf ",s(a,b,c));**

**else printf("%.0lf ",s(a,b,c));**

**printf("%.6lf\n",area(a,b,c));**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：a=b=c=4;a=b=c=2；a=6，b=8, c=10;

（b） 对应测试运行结果如图所示。

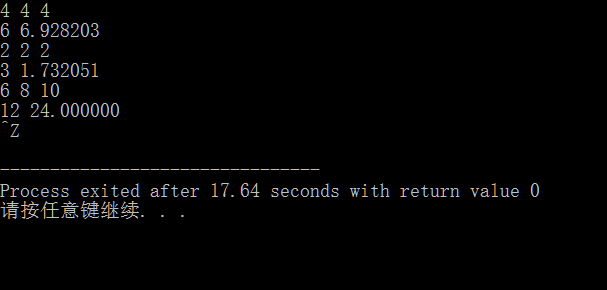


图4-1 编程题1的测试用例一的运行结果

（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。

**1）解题思路**:

1.宏定义CHANGE

2.以字符串形式读入一整行，存入字符数组

3.利用CHANGE判断输出方式

4. 输出结果

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**#define CHANGE c[0]%2**

**char c[1005];**

**int n;**

**int main(){**

**scanf("%d",&n);getchar();**

**for(int i=1;i<=n;i++){**

**fgets(c,1005,stdin);**

**if(CHANGE){**

**for(int j=0;j<strlen(c);j++){**

**if(c[j]>='a'&&c[j]<='z') printf("%c",c[j]-32);**

**else if(c[j]>='A'&&c[j]<='Z') printf("%c",c[j]+32);**

**else printf("%c",c[j]);**

**}**

**}**

**else{**

**for(int j=0;j<strlen(c);j++)**

**printf("%c",c[j]);**

**}**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：见运行结果图

（b） 对应测试运行结果如图所示。

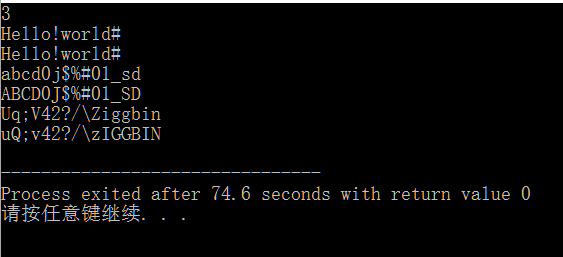


图4-2 编程题2的测试用例一的运行结果

## 4.3 实验小结

在实验中我掌握了宏定义的用法，以及一些编译预处理的知识。采用宏可以大大简化编程复杂度。

# 5 数组实验

## 5.1 实验目的

（1）掌握数组的说明、初始化和使用。

（2）掌握一维数组作为函数参数时实参和形参的用法。

（3）掌握字符串处理函数的设计，包括串操作函数及数字串与数之间转换函数实现算法。

（4）掌握基于分治策略的二分查找算法和选择法排序算法的思想，以及相关算法的实现。

## 5.2 实验内容及要求

**5.2.1 源程序改错**

下面是用来将数组a中元素按升序排序后输出的源程序。分析源程序中存在的问题，并对源程序进行修改，使之能够正确完成任务。

源程序

1 #include<stdio.h>

2 int main(void)

3 {

4 int a[10] = {27, 13, 5, 32, 23, 3, 17, 43, 55, 39};

5 void sort(int [],int);

6 int i;

7 sort(a[0],10);

8 for(i = 0; i < 10; i++)

9 printf("%6d",a[i]);

10 printf("\n");

11 return 0;

12 }

13 void sort(int b[], int n)

14 {

15 int i, j, t;

16 for (i = 0; i < n - 1; i++)

17 for ( j = 0; j < n - i - 1; j++)

18 if(b[j] < b[j+1])

19 t = b[j], b[j] = b[j+1], b[j+1] = t;

20 }

**解答：**

（1）错误修改：

1) 第5行函数声明应在main函数外。

2) 第7行函数参数应为数组名，正确形式为：**sort(a[0],10);**

3) 第18行升序输出应为>号，正确形式为：**if(b[j] >b[j+1])**

5.2.2 源程序完善、修改、替换

(1) 下面的源程序用于求解约瑟夫问题：M个人围成一圈，从第一个人开始依次从1至N循环报数，每当报数为N时报数人出圈，直到圈中只剩下一个人为止。请在源程序中的下划线处填写合适的代码来完善该程序。

源程序：

#include<stdio.h>

#define M 10

#define N 3

int main(void)

{

int a[M], b[M]; /\* 数组a存放圈中人的编号，数组b存放出圈人的编号 \*/

int i, j, k;

for(i = 0; i < M; i++) /\* 对圈中人按顺序编号1—M \*/

a[i] = i + 1;

for(i = M, j = 0; i > 1; i--){

/\* i表示圈中人个数，初始为M个，剩1个人时结束循环；j表示当前报数人的位置 \*/

for(k = 1; k <= N; k++) /\* 1至N报数 \*/

if(++j > i - 1) j = 0;/\* 最后一个人报数后第一个人接着报，形成一个圈 \*/

b[M-i] = j? a[j-1] : a[i-1] ; /\* 将报数为N的人的编号存入数组b \*/

if(j)

for(k = --j; k < i; k++) /\* 压缩数组a，使报数为N的人出圈 \*/

a[k]=a[k+1] ;

}

for(i = 0;i < M – 1; i++) /\* 按次序输出出圈人的编号 \*/

printf(“%6d”, b[i]);

printf(“%6d\n”, a[0]); /\* 输出圈中最后一个人的编号 \*/

return 0;

}

(2) 上面的程序中使用数组元素的值表示圈中人的编号，故每当有人出圈时都要压缩数组，这种算法不够精炼。如果采用做标记的办法，即每当有人出圈时对相应数组元素做标记，从而可省掉压缩数组的时间，这样处理效率会更高一些。因此，请采用做标记的办法修改（1）中的程序，并使修改后的程序与（1）中的程序具有相同的功能。

#include<stdio.h>

#define M 10

#define N 3

int vis[M]; //用来做标记的数组

int main(void)

{

int a[M], b[M]; /\* 数组a存放圈中人的编号，数组b存放出圈人的编号 \*/

int i, j, k;

for(i = 0; i < M; i++) /\* 对圈中人按顺序编号1-M \*/

a[i] = i + 1;

for(i = M, j = 0; i > 1; i--){

/\* i表示圈中人个数，初始为M个，剩1个人时结束循环；j表示当前报数人的位置 \*/

for(k = 1; k < N; k++) /\* 1至N报数 \*/

do{

j++;

if(j > M - 1)

j = 0; /\* 最后一个人报数后第一个人接着报，形成一个圈 \*/

}while(vis[j]);

vis[j]=1;

b[M-i] =a[j]; /\* 将报数为N的人的编号存入数组b \*/

do{

j++;

if(j > M - 1)

j = 0;

}while(vis[j]);

}

for(i = 0;i < M - 1; i++) /\* 按次序输出出圈人的编号 \*/

printf("%6d", b[i]);

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

if(!vis[i]){

printf("%6d\n", a[i]);

break;

}

/\* 输出圈中最后一个人的编号 \*/

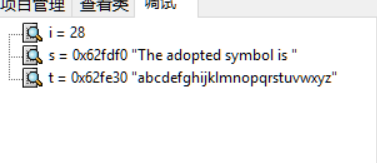
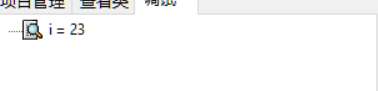
return 0;

}

5.2.3 跟踪调试源程序

在下面所给的源程序中，函数strncat(s,t,n)本来应该将字符数组t的前n个字符连接到字符数组s中字符串的尾部。但函数strncat在定义时代码有误，不能实现上述功能。请按下面的要求进行操作，并回答问题和排除错误。

（1） 单步执行源程序。进入函数strncat后观察表达式s、t和i。当光条落在for语句所在行时，i为何值？当光条落在strncat函数块结束标记（右花括号 }）所在行时, s、t分别为何值？



（2）分析函数出错的原因，排除错误，使函数正确实现功能，最后写出程序的输出结果。在strncat函数中的for循环语句前加一条i--;语句即可。

源程序：

#include<stdio.h>

void strncat(char [],char [],int);

int main(void)

{

char a[50]="The adopted symbol is ",b[27]="abcdefghijklmnopqrstuvwxyz";

strncat(a, b, 4);

printf("%s\n",a);

return 0;

}

void strncat(char s[],char t[], int n)

{

int i = 0, j;

while(s[i++]) ;

for(j = 0; j < n && t[j];)

s[i++] = t[j++];

s[i] = '\0';

}

5.2.4 程序设计

1. 编写一个程序,从键盘读取数据，对一个3×4矩阵进行赋值，求其转置矩阵，然后输出原矩阵和转置矩阵。
2. **解题思路:**

基本数组赋值运算

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int matrx[5][5],ans[5][5];**

**int main(){**

**for(int i=1;i<=3;i++)**

**for(int j=1;j<=4;j++)**

**scanf("%d",&matrx[i][j]);**

**for(int i=1;i<=3;i++){**

**for(int j=1;j<=4;j++)**

**printf("%5d",matrx[i][j]);**

**printf("\n");**

**}**

**printf("\n");**

**for(int i=1;i<=4;i++){**

**for(int j=1;j<=3;j++){**

**ans[i][j]=matrx[j][i];**

**printf("%5d",ans[i][j]);**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：1 2 3 4

5 6 7 8

9 10 11 12

（b） 对应测试运行结果如图所示。

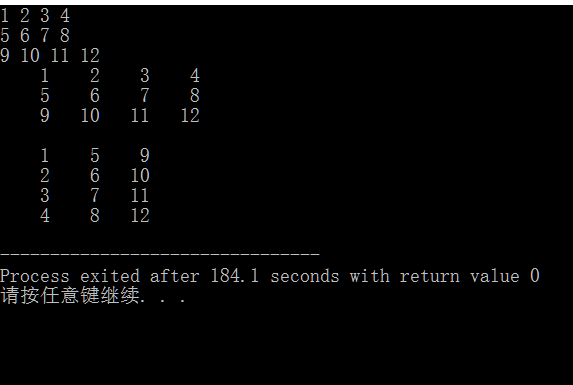


图5-1 编程题1的测试用例一的运行结果

1. 编写一个程序, 其功能要求是：输入一个整数，将它在内存中二进制表示的每一位转换成为对应的数字字符，存放到一个字符数组中，然后输出该整数的二进制表示。

1）**解题思路:**

1.利用sizeof()判断机器字长

2.按照不同字长用位运算取出整数的每一位

3.将每一位放进数组中输出

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int array[35],n,a;**

**int main(){**

**scanf("%d",&n);**

**for(int i=1;i<=n;i++){**

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

**if(sizeof(int)==4){**

**unsigned int temp=(unsigned)a;**

**for(int j=32;j>=1;j--){**

**array[j]=temp&1;**

**temp>>=1;**

**}**

**for(int j=1;j<=32;j++)**

**printf("%d",array[j]);**

**printf("\n");**

**}**

**else{**

**unsigned int temp=(unsigned)a;**

**for(int j=16;j>=1;j--){**

**array[j]=temp&1;**

**temp>>=1;**

**}**

**for(int j=1;j<=16;j++)**

**printf("%d",array[j]);**

**printf("\n");**

**}**

**}**

**}**

3）**测试**

（a） 测试数据：8

1 1024 26 0 -1 65 4 -1024

（b） 对应测试运行结果如图所示。

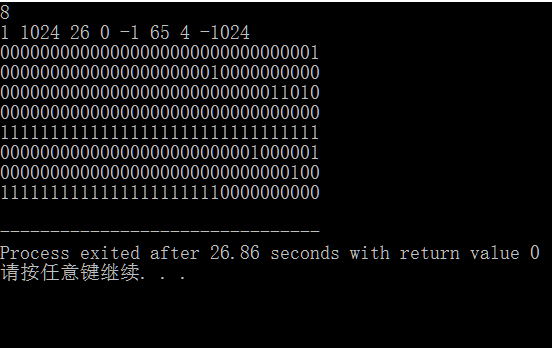


图5-2 编程题2的测试用例一的运行结果

1. 编写一个程序, 其功能要求是：输入n个学生的姓名和C语言课程的成绩，将成绩按从高到低的次序排序，姓名同时作相应调整，输出排序后学生的姓名和C语言课程的成绩。然后，输入一个C语言课程成绩值，用二分查找进行搜索。如果查找到有该成绩，输出该成绩同学的姓名和C语言课程的成绩；否则输出提示“not found!”。
   1. **解题思路:**

1.自定义结构体，封装成绩与名字

2.自定义结构体之间的比较函数cmp()

3.利用qsort()完成排序

4.递归实现二分查找

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**#include<stdbool.h>**

**//#include<algorithm>**

**//using namespace std;**

**int n,m;**

**struct person{**

**char name[21];**

**int score;**

**}student[102];**

**int cmp(const void\* a,const void \* b){**

**return (\*(struct person\*)a).score<(\*(struct person\*)b).score;**

**}**

**int find(int head,int last,int val){**

**if(head>last) return -1;**

**int mid=(head+last)/2;**

**if(student[mid].score>val) return find(mid+1,last,val);**

**if(student[mid].score<val) return find(head,mid-1,val);**

**return mid;**

**}**

**int main(){**

**scanf("%d",&n);**

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

**scanf("%s%d",student[i].name,&student[i].score);**

**qsort(student,n,sizeof(student[0]),cmp);**

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

**printf("%-20s %d\n",student[i].name,student[i].score);**

**printf("\n");**

**scanf("%d",&m);**

**for(int i=1;i<=m;i++){**

**int a;**

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

**if((a=find(0,n-1,a))!=-1)printf("%20s %d\n",student[a].name,student[a].score);**

**else printf("Not found!\n");**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：

5

ZhangChuanChao 88

XiaoHong 95

XiaoMing 90

LiSi 100

XiDaDa 83

4

90 88 83 59

（b） 对应测试运行结果如图所示。

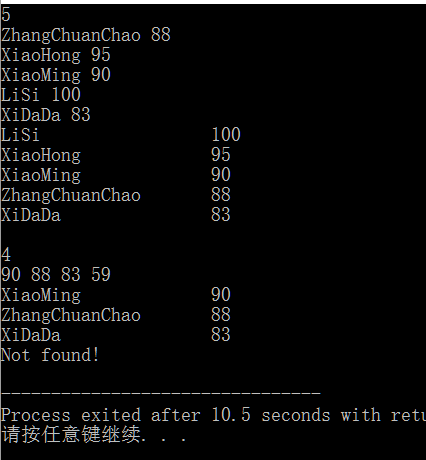


图5-3 编程题3的测试用例一的运行结果

5.2.5 选做题

1. 编写函数strnins（s，t，n），其功能是：可将字符数组t中的字符串插入到字符数组s中字符串的第n个字符的后面。

1）**解题思路**:

1.判断n与s长度的关系

2.将n后面的字符向后推

3.将t字符串插入。

2）**源代码**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**char a[100],b[100];**

**int n;**

**void strnins(char s[],char t[] ,int n){**

**int len=strlen(t);**

**if(n>strlen(s)){**

**s[strlen(s)]=' ';**

**int i;**

**for(i=n;i<=n+len;i++)**

**s[i]=t[i];**

**s[i]='\0';**

**return ;**

**}**

**for(int i=n;i<strlen(s);i++)**

**s[i+len]=s[i];**

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

**s[n+i]=t[i];**

**}**

**int main(){**

**scanf("%d",&n);**

**scanf("%s%s",a,b);**

**strnins(a,b,n);**

**printf("%s\n",a);**

**return 0;**

**}**

1. 编写一个实现八皇后问题的程序，即：在8\*8方格国际象棋棋盘上放置8个皇后，任意两个皇后不能位于同一行、同一列或同一斜线上（正斜线或反斜线），并输出所有可能的放法。

1）**解题思路**:

1. 深度优先搜索逐行放置，回溯判断，具体见代码。

2）**源代码**

/**/#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**#define N 8**

**int map[10][10];**

**int ans[10][10];**

**int ansnum;**

**int cnt;**

**void placequeen(int x,int y,int num){**

**ans[x][y]=num;**

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

**for(int j=1;j<=N;j++)**

**if(abs(i-x)==abs(j-y))**

**if(map[i][j]==0) map[i][j]=num;**

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

**if(!map[i][y]) map[i][y]=num;**

**for(int j=1;j<=N;j++)**

**if(!map[x][j]) map[x][j]=num;**

**}**

**void clearqueen(int x,int y,int num){**

**ans[x][y]=0;**

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

**for(int j=1;j<=N;j++)**

**if(abs(i-x)==abs(j-y))**

**if(map[i][j]==num) map[i][j]=0;**

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

**if(map[i][y]==num) map[i][y]=0;**

**for(int j=1;j<=N;j++)**

**if(map[x][j]==num) map[x][j]=0;**

**}**

**void printans(){**

**printf("This is the %d answer:\n",cnt);**

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

**for(int j=1;j<=N;j++)**

**if(ans[i][j]==0) printf("\*");**

**else printf("%d",ans[i][j]);**

**printf("\n");**

**}**

**printf("\n");**

**}**

**void eightqueen(int have\_done){**

**if(have\_done==N){**

**cnt++;**

**printans();**

**return ;**

**}**

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

**if(map[have\_done+1][i]==0){**

**placequeen(have\_done+1,i,have\_done+1);**

**eightqueen(have\_done+1);**

**clearqueen(have\_done+1,i,have\_done+1);**

**}**

**}**

**int main(){**

**eightqueen(0);**

**return 0;**

**}**

共92组解，详情见代码。

## 5.3 实验小结

在这次实验中，我掌握了数组的使用方法和递归的写法，尤其多维指针使我加深了对内存的理解，八皇后问题锻炼了我的代码能力。

# 6 指针实验

## 6.1 实验目的

(1)熟练掌握指针的说明、赋值、使用。

(2)掌握用指针引用数组的元素，熟悉指向数组的指针的使用。

(3)熟练掌握字符数组与字符串的使用，掌握指针数组及字符指针数组的用法。

(4)掌握指针函数与函数指针的用法。

(5)掌握带有参数的main函数的用法。

## 6.2 实验内容

**6.2.1源程序改错题**

下面程序是否存在错误？如果存在，原因是什么？如果存在错误，要求在计算机上对这个例子程序进行调试修改，使之能够正确执行。

1 #include "stdio.h"

2 void main(void)

3 {

4 float \*p;

5 scanf("%f",p);

6 printf("%f\n",\*p);

7 }

**解答：**

（1）错误修改：

1) 第3行为读入的float型申请内存，添加：**float a;**

2) 第4行指针为空指针，应为其赋值：**float \*p=&a**；

**6.2.2源程序完善、修改、替换题**

（1）下面的程序通过函数指针和菜单选择来调用字符串拷贝函数或字符串连接函数，请在下划线处填写合适的表达式、语句、或代码片段来完善该程序。

#include "stdio.h"

#include "string.h"

void main(void)

{

char \*(\*p)(char \*a,const char \*b);

char a[80],b[80],c[160],\*result=c;

int choice,i;

do{

printf("\t\t1 copy string.\n");

printf("\t\t2 connect string.\n");

printf("\t\t3 exit.\n");

printf("\t\tinput a number (1-3) please!\n");

scanf("%d",&choice);

}while(choice<1 || choice>5);

switch(choice){

case 1:

p=strcpy;

break;

case 2:

p=strcat;

break;

case 3:

goto down;

}

getchar();

printf("input the first string please!\n");

i=0;

while(scanf("%c",&a[i]),a[i]!='\n') i++; a[i]='\0';

printf("input the second string please!\n");

i=0;

while(scanf("%c",&a[i]),a[i]!='\n') i++; a[i]='\0';

result= p (a,b);

printf("the result is %s\n",result);

down:

;

}

（2）为了使程序不受scanf、getchar、gets等函数输入后回车符的影响，请修改第（1）题程序，按要求输出下面结果：（（输入）表示该数据是键盘输入数据）

1 copy string.

2 connect string.

3 exit.

input a number (1-3) please!

2 （输入）

input the first string please!

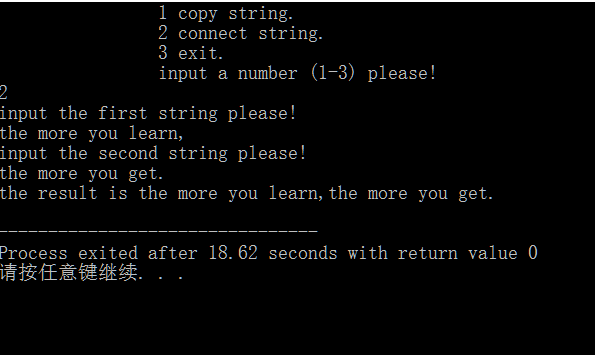
the more you learn, （输入）

input the second string please!

the more you get. （输入）

the result is the more you learn,the more you get.

上题所示程序即可完成任务，结果截图如下：



**6.2.3跟踪调试题**

#include "stdio.h"

char \*strcpy(char \*,char \*);

void main(void)

{

char a[20],b[60]="there is a boat on the lake.";

printf("%s\n",strcpy(a,b));

}

char \*strcpy(char \*s,char \*t)

{

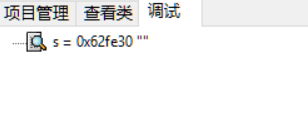
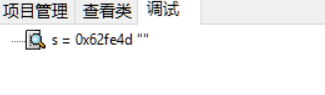
while(\*s++=\*t++)

;

return (s);

}

1. 单步执行。进入strcpy时watch窗口中s为何值？返回main时, watch窗口中s为何值？

进入时：返回时：

（2）排除错误，使程序输出结果为：

there is a boat on the lake.

修改为如下：

#include "stdio.h"

char \*strcpy(char \*,char \*);

int main(void)

{

char a[60],b[60]="there is a boat on the lake.";

strcpy(a,b);

printf("%s\n",a);

}

char \*strcpy(char \*s,char \*t)

{

while(\* s++ =\* t++);

return (s);

}

**6.2.4编程设计题**

（1）一个长整型变量占4个字节，其中每个字节又分成高4位和低4位。试从该长整型变量的高字节开始，依次取出每个字节的高4位和低4位并以数字字符的形式进行显示。

1. **解题思路:**

制作mask后利用位运算依次取出每个字节的高低位，输出。

2）**源程序**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int n,a[30];**

**void sol(unsigned int x){**

**unsigned mask=0xf0000000;**

**//printf("%x\n",mask);**

**printf("%X ",(mask&x)>>28);**

**for(int i=1;i<=6;i++){**

**mask>>=4;**

**printf("%X ",(mask&x)>>(28-4\*i));**

**}**

**mask>>=4;**

**printf("%X\n",mask&x);**

**}**

**int main(){**

**scanf("%d",&n);**

**for(int i=1;i<=n;i++)**

**scanf("%d",a+i);**

**for(int i=1;i<=n;i++)//对于数组中每个数依次处理**

**sol((unsigned)\*(a+i));**

**return 0;**

**}**

3）**测试**

（a） 测试数据：6

1 0 88888 90 -23 10

（b） 对应测试运行结果如图所示。

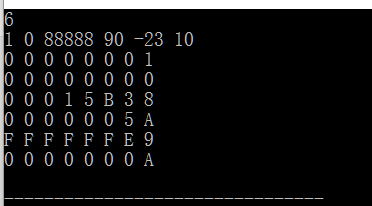


图6-1 编程题1的测试用例一的运行结果

（2）利用大小为n的指针数组指向用gets函数输入的n行，每行不超过80个字符。编写一个函数，它将每一行中连续的多个空格字符压缩为一个空格字符。在调用函数中输出压缩空格后的各行，空行不予输出。

1. **解题思路:**

对于每个字符串扫描，记录空格数，如果空格数为0则输出空格，否则忽略。

2）**源程序**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**char s[100][85];**

**char \*p[100];**

**int n;**

**void sol(char \*x){**

**int space\_num=0;**

**while((\*x)!='\n'){**

**if((\*x)!=' ') {**

**space\_num=0;**

**printf("%c",\*x);**

**}**

**else{**

**if(space\_num==0){**

**space\_num++;**

**printf("%c",\*x);**

**}**

**}**

**x++;**

**}**

**printf("\n");**

**}**

**int main(){**

**while(scanf("%d",&n),n){**

**getchar();**

**for(int i=1;i<=n;i++)//利用fget()读入**

**fgets(s[i],85,stdin);**

**for(int i=1;i<=n;i++)**

**p[i]=s[i];**

**for(int i=1;i<=n;i++)**

**if((\*p[i])!='\n')**

**sol(p[i]);**

**printf("\n");**

**}**

**return 0;**

**}**

3）**测试**

（a） 测试数据：3

zhang chuan chao

xi da da

4

If only we'd stop trying to be happy

we'd have a pretty good time.

0

（b） 对应测试运行结果如图所示。

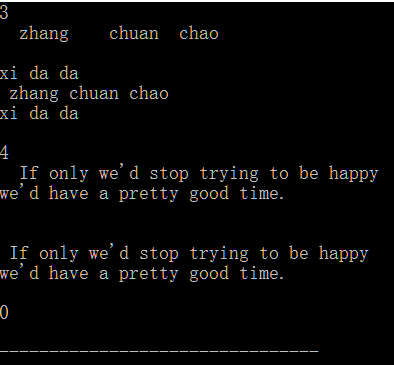


图6-2 编程题2的测试用例一的运行结果

（3）设某个班有N个学生，每个学生修了M门课程（用#define定义N、M）。输入M门课程的名称，然后依次输入N个学生中每个学生所修的M门课程的成绩并且都存放到相应的数组中。编写下列函数：

a.计算每个学生各门课程平均成绩；

b.计算全班每门课程的平均成绩；

c.分别统计低于全班各门课程平均成绩的人数；

d.分别统计全班各门课程不及格的人数和90分以上（含90分）的人数。

在调用函数中输出上面各函数的计算结果。（要求都用指针操作，不得使用下标操作。）

1. **解题思路:**

将名字与成绩定义在一个结构中，声明结构数组和指向其的指针数组，每个功能对应一个函数，在函数中用指针访问数据，完成操作。

2）**源程序**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**#define N 5**

**#define M 5**

**char subject[N][20];**

**struct student{**

**char name[20];**

**int score[N];**

**}stu[M],\*p[M];**

**double ave[N];**

**void get\_student\_avescore(struct student \*x){**

**double ans=0.0;**

**for(int i=0;i<N;i++)**

**ans+=(double)(x->score[i]);**

**printf("Average score of %s is %.2lf\n",x->name,ans/N);**

**}**

**void get\_subject\_avescore(int x){**

**double ans=0.0;**

**for(int i=0;i<M;i++)**

**ans+=(double)(p[i]->score[x]);**

**printf("Average score of %s is %.2lf\n",\*(subject+x),\*(ave+x)=ans/M);**

**}**

**void get\_under\_ave\_num(int x){**

**int num=0;**

**for(int i=0;i<M;i++)**

**if((double)(p[i]->score[x])<ave[x]) num++;**

**printf("Number of students lower than avg of %s is %d\n",\*(subject+x),num);**

**}**

**void get\_fail\_num(int x){**

**int num=0;**

**for(int i=0;i<N;i++)**

**if(p[i]->score[x]<60) num++;**

**printf("Number of students %s fail is %d\n",\*(subject+x),num);**

**}**

**void get\_perfect\_num(int x){**

**int num=0;**

**for(int i=0;i<N;i++)**

**if(p[i]->score[x]>=90) num++;**

**printf("Number of students %s perfect is %d\n",\*(subject+x),num);**

**}**

**int main(){**

**for(int i=0;i<M;i++)**

**p[i]=&stu[i];**

**for(int i=0;i<N;i++)**

**scanf("%s",\*(subject+i));**

**for(int i=0;i<M;i++){**

**scanf("%s",stu[i].name);**

**for(int j=0;j<N;j++){**

**scanf("%d",stu[i].score+j);**

**}**

**}**

**for(int i=0;i<M;i++)**

**get\_student\_avescore(p[i]);**

**for(int i=0;i<N;i++)**

**get\_subject\_avescore(i);**

**for(int i=0;i<N;i++)**

**get\_under\_ave\_num(i);**

**for(int i=0;i<N;i++)**

**get\_fail\_num(i);**

**for(int i=0;i<N;i++)**

**get\_perfect\_num(i);**

**return 0;**

**}**

3）**测试**

（a） 测试数据：A B C D E

Zhang

87 88 77 87 95

Li

88 98 100 48 75

Wang

85 68 95 47 59

Han

86 89 75 85 88

Gan

87 68 87 89 100

（b） 对应测试运行结果如图所示。

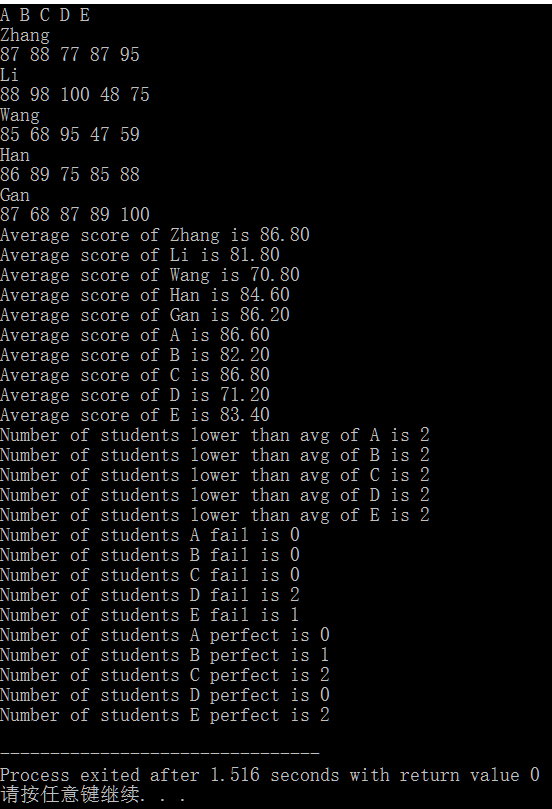


图6-3 编程题3的测试用例一的运行结果

（4）编写一个程序，输入n个数，排序后输出。排序的原则是由命令行可选参数-d决定，并且有参数-d时按递减顺序排序，否则按递增顺序排序。要求将排序算法定义成函数，利用指向函数的指针使该函数实现递增或递减排序。

1. **解题思路:**

判断命令行参数，依据参数个数及内容给排序函数传入不同参数

2）**源程序**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**int n,a[100];**

**void swap(int \*a,int \*b){**

**int t=\*a;**

**\*a=\*b;**

**\*b=t;**

**}**

**void sort(int \*b,int flag){**

**if(flag){**

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

**for(int j=i+1;j<n;j++)**

**if(b[i]<b[j]) swap(&b[i],&b[j]);**

**}**

**else for(int i=0;i<n;i++)**

**for(int j=i+1;j<n;j++)**

**if(b[i]>b[j]) swap(&b[i],&b[j]);**

**}**

**int main(int argc,char \*argv[]){**

**scanf("%d",&n);**

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

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

**void (\*p)(int \*b,int x);**

**if(argc>1&&strcmp(argv[1],"-d"))//判断是否接收到-d参数**

**p(a,1);**

**else p(a,0);**

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

**printf("%d ",a[i]);**

**printf("\n");**

**return 0;**

**}**

## 6.3 实验小结

在这次实验中，我知道了指针的使用方法，了解了间接访问与直接访问，掌握了命令行参数的使用，编程能力大幅提高

# 7 结构与联合实验

## 7.1 实验目的

(1)通过实验，熟悉和掌握结构的说明和引用、结构的指针、结构数组、以及函数中使用结构的方法。

(2)通过实验，掌握动态储存分配函数的用法，掌握自引用结构，单向链表的创建、遍历、结点的增删、查找等操作。

(3)了解字段结构和联合的用法。

## 7.2 实验内容

**7.2.1表达式求值的程序验证题**

设有说明：

char u[]="UVWXYZ";

char v[]="xyz";

struct T{

int x;

char c;

char \*t;

}a[]={{11,ˊAˊ,u},{100, ˊBˊ,v}},\*p=a;

请先自己计算下面表达式的值，然后通过编程计算来加以验证。(各表达式相互无关)

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **表达式** | **计算值** | **验证值** |
| 1 | (++p)->x | 100 | 100 |
| 2 | p++,p->c | B | B |
| 3 | \*p++->t,\*p->t | x | x |
| 4 | \*(++p)->t | x | x |
| 5 | \*++p->t | V | V |
| 6 | ++\*p->t | V | V |

**7.2.2源程序修改替换题**

给定一批整数，以0作为结束标志且不作为结点，将其建成一个先进先出的链表，先进先出链表的头指针始终指向最先创建的结点（链头），先建结点指向后建结点，后建结点始终是尾结点。

1. 源程序中存在什么样的错误（先观察执行结果）？对程序进行修改、调试，使之能够正确完成指定任务。

源程序如下：

#include "stdio.h"

#include "stdlib.h"

struct s\_list{

int data; /\* 数据域 \*/

struct s\_list \*next; /\* 指针域 \*/

} ;

void create\_list (struct s\_list \*headp,int \*p);

void main(void)

{

struct s\_list \*head=NULL,\*p;

int s[]={1,2,3,4,5,6,7,8,0}; /\* 0为结束标记 \*/

create\_list(head,s); /\* 创建新链表 \*/

p=head; /\*遍历指针p指向链头 \*/

while(p){

printf("%d\t",p->data); /\* 输出数据域的值 \*/

p=p->next; /\*遍历指针p指向下一结点 \*/

}

printf("\n");

}

void create\_list(struct s\_list \*headp,int \*p)

{

struct s\_list \* loc\_head=NULL,\*tail;

if(p[0]==0) /\* 相当于\*p==0 \*/

;

else { /\* loc\_head指向动态分配的第一个结点 \*/

loc\_head=(struct s\_list \*)malloc(sizeof(struct s\_list));

loc\_head->data=\*p++; /\* 对数据域赋值 \*/

tail=loc\_head; /\* tail指向第一个结点 \*/

while(\*p){ /\* tail所指结点的指针域指向动态创建的结点 \*/

tail->next=(struct s\_list \*)malloc(sizeof(struct s\_list));

tail=tail->next; /\* tail指向新创建的结点 \*/

tail->data=\*p++; /\* 向新创建的结点的数据域赋值 \*/

}

tail->next=NULL; /\* 对指针域赋NULL值 \*/

}

headp=loc\_head; /\* 使头指针headp指向新创建的链表 \*/

}

**答：将第7、12、20行分别修改如下即可。**

**第7行修改为void create\_list (struct s\_list \*\*headp,int \*p);**

**第12行修改为create\_list(&head,s);**

**第20行修改为void create\_list(struct s\_list \*\*headp,int \*p)**

（2）修改替换create\_list函数，将其建成一个后进先出的链表，后进先出链表的头指针始终指向最后创建的结点（链头），后建结点指向先建结点，先建结点始终是尾结点。

**答：修改creat\_list函数如下：**

**void create\_list(struct s\_list \*\*headp,int \*p)**

**{**

**struct s\_list \* loc\_head=NULL,\*tail;**

**if(p[0]==0) /\* 相当于\*p==0 \*/**

**;**

**else { /\* loc\_head指向动态分配的第一个结点 \*/**

**loc\_head=(struct s\_list \*)malloc(sizeof(struct s\_list));**

**loc\_head->data=\*p++; /\* 对数据域赋值 \*/**

**tail=loc\_head; /\* tail指向第一个结点 \*/**

**tail->next=NULL;**

**while(\*p){ /\* tail所指结点的指针域指向动态创建的结点 \*/**

**tail=(struct s\_list \*)malloc(sizeof(struct s\_list));**

**tail->next=loc\_head; /\* tail指向新创建的结点 \*/**

**tail->data=\*p++; /\* 向新创建的结点的数据域赋值 \*/**

**loc\_head=tail;**

**}**

**}**

**\*headp=tail; /\* 使头指针headp指向新创建的链表 \*/**

**}**

**7.2.3编程设计题**

（1）设计一个字段结构struct bits，它将一个8位无符号字节从最低位向最高位声明为8个字段，各字段依次为bit0, bit1, …, bit7，且bit0的优先级最高。同时设计8个函数，第i个函数以biti(i=0,1,2,…,7)为参数，并且在函数体内输出biti的值。将8个函数的名字存入一个函数指针数组p\_fun。如果bit0为1，调用p\_fun[0]指向的函数。如果struct bits中有多位为1，则根据优先级从高到低依次调用函数指针数组p\_fun中相应元素指向的函数。8个函数中的第0个函数可以设计为：

void f0(struct bits b)

{

Printf(“the function %d is called!\n”,b);

}

**1)解题思路**:

按照题意定义含八个字段的结构体以及函数指针数组，根据输入为字段赋值，根据字段值调用相关函数输出。

2)**源程序**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**struct ISR\_BITS{**

**unsigned int bits0:1;**

**unsigned int bits1:1;**

**unsigned int bits2:1;**

**unsigned int bits3:1;**

**unsigned int bits4:1;**

**unsigned int bits5:1;**

**unsigned int bits6:1;**

**unsigned int bits7:1;**

**};**

**union ISR\_REG{**

**unsigned short all;**

**struct ISR\_BITS bit;**

**}bit\_now;**

**int num[100],n;**

**void (\*p\_isr[8])(void);**

**//题意中的中断函数**

**void isr0(){**

**printf("The Interrupt Service Routine isr0 is called!\n");**

**}**

**void isr1(){**

**printf("The Interrupt Service Routine isr1 is called!\n");**

**}**

**void isr2(){**

**printf("The Interrupt Service Routine isr2 is called!\n");**

**}**

**void isr3(){**

**printf("The Interrupt Service Routine isr3 is called!\n");**

**}**

**void isr4(){**

**printf("The Interrupt Service Routine isr4 is called!\n");**

**}**

**void isr5(){**

**printf("The Interrupt Service Routine isr5 is called!\n");**

**}**

**void isr6(){**

**printf("The Interrupt Service Routine isr6 is called!\n");**

**}**

**void isr7(){**

**printf("The Interrupt Service Routine isr7 is called!\n");**

**}**

**int main(){**

**p\_isr[0]=isr0;p\_isr[1]=isr1;p\_isr[2]=isr2;p\_isr[3]=isr3;**

**p\_isr[4]=isr4;p\_isr[5]=isr5;p\_isr[6]=isr6;p\_isr[7]=isr7;**

**scanf("%d",&n);**

**for(int i=1;i<=n;i++)**

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

**for(int i=1;i<=n;i++){**

**printf("%d:\n",num[i]);**

**for(int j=0;j<1;j++){**

**if(num[i]&1) bit\_now.bit.bits0=1;**

**else bit\_now.bit.bits0=0;**

**num[i]>>=1;**

**}**

**for(int j=1;j<2;j++){**

**if(num[i]&1) bit\_now.bit.bits1=1;**

**else bit\_now.bit.bits1=0;**

**num[i]>>=1;**

**}**

**for(int j=2;j<3;j++){**

**if(num[i]&1) bit\_now.bit.bits2=1;**

**else bit\_now.bit.bits2=0;**

**num[i]>>=1;**

**}**

**for(int j=3;j<4;j++){**

**if(num[i]&1) bit\_now.bit.bits3=1;**

**else bit\_now.bit.bits3=0;**

**num[i]>>=1;**

**}**

**for(int j=4;j<5;j++){**

**if(num[i]&1) bit\_now.bit.bits4=1;**

**else bit\_now.bit.bits4=0;**

**num[i]>>=1;**

**}**

**for(int j=5;j<6;j++){**

**if(num[i]&1) bit\_now.bit.bits5=1;**

**else bit\_now.bit.bits5=0;**

**num[i]>>=1;**

**}**

**for(int j=6;j<7;j++){**

**if(num[i]&1) bit\_now.bit.bits6=1;**

**else bit\_now.bit.bits6=0;**

**num[i]>>=1;**

**}**

**for(int j=7;j<8;j++){**

**if(num[i]&1) bit\_now.bit.bits7=1;**

**else bit\_now.bit.bits7=0;**

**num[i]>>=1;**

**}**

**if(bit\_now.bit.bits0==1) p\_isr[0]();**

**if(bit\_now.bit.bits1==1) p\_isr[1]();**

**if(bit\_now.bit.bits2==1) p\_isr[2]();**

**if(bit\_now.bit.bits3==1) p\_isr[3]();**

**if(bit\_now.bit.bits4==1) p\_isr[4]();**

**if(bit\_now.bit.bits5==1) p\_isr[5]();**

**if(bit\_now.bit.bits6==1) p\_isr[6]();**

**if(bit\_now.bit.bits7==1) p\_isr[7]();**

**printf("\n");**

**}**

**return 0;**

}

3）**测试**

（a） 测试数据：5

56 0 255 8 89

（b） 对应测试运行结果如图所示。

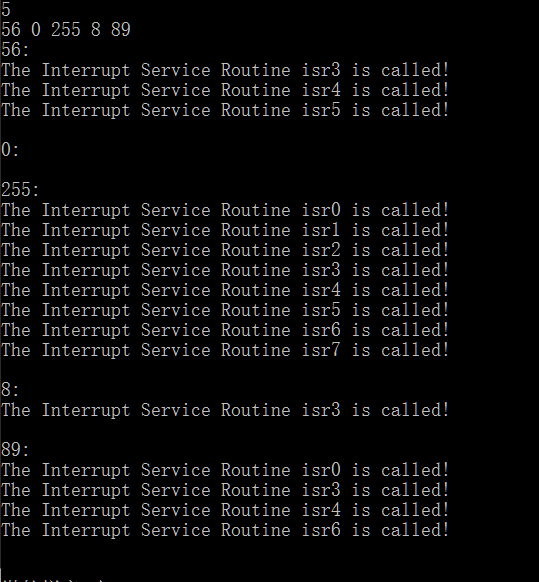


图7-1 编程题1的测试用例一的运行结果

（2）用单向链表建立一张班级成绩单，包括每个学生的学号、姓名、英语、高等数学、普通物理、C语言程序设计四门课程的成绩。用函数编程实现下列功能：

(1) 输入每个学生的各项信息。

(2) 输出每个学生的各项信息。

(3) 修改指定学生的指定数据项的内容。

(4) 统计每个同学的平均成绩（保留2位小数）。

(5) 输出各位同学的学号、姓名、四门课程的总成绩和平均成绩。

1. **解题思路:**

定义学生类型，包括题中信息及平均成绩，用链表存储结构信息，完成题中所给操作，以链表形式输出。

2）**源程序**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**struct Student{**

**char num[15];**

**char name[20];**

**float math,physics,english,c;**

**float sum,avg;**

**struct Student \*next;**

**};**

**int n,m;**

**void Studentcopy(struct Student \*a,struct Student \*b){**

**strcpy(b->num,a->num);**

**strcpy(b->name,a->name);**

**b->avg=a->avg;**

**b->sum=a->sum;**

**b->english=a->english;**

**b->math=a->math;**

**b->physics=a->physics;**

**b->c=a->c;**

**}**

**int main(){**

**scanf("%d",&n);**

**struct Student \*head=(struct Student \*)malloc(sizeof(struct Student));**

**struct Student \*tail=head;**

**scanf("%s%s%f%f%f%f",head->num,head->name,&head->english,&head->math,&head->physics,&head->c);**

**for(int i=1;i<n;i++){**

**tail->next=(struct Student \*)malloc(sizeof(struct Student));**

**tail=tail->next;**

**scanf("%s%s%f%f%f%f",tail->num,tail->name,&tail->english,&tail->math,&tail->physics,&tail->c);**

**tail->next=NULL;**

**}**

**printf("%-15s%-20s%-10s%-10s%-10s%-10s\n","ID","Name","English","Math","Physics","C");**

**tail=head;**

**for(;tail!=NULL;tail=tail->next){**

**printf("%-15s%-20s%-10.2f%-10.2f%-10.2f%-10.2f\n",tail->num,tail->name,tail->english,tail->math,tail->physics,tail->c);**

**}**

**printf("\n");**

**scanf("%d",&m);**

**for(int i=1;i<=m;i++){**

**char temp\_num[15],subject[10];**

**float new\_score;**

**scanf("%s%s%f",temp\_num,subject,&new\_score);**

**tail=head;**

**for(;tail!=NULL;tail=tail->next)**

**if(strcmp(tail->num,temp\_num)==0) break;**

**if(strcmp(subject,"English")==0) tail->english=new\_score;**

**if(strcmp(subject,"Math")==0) tail->math=new\_score;**

**if(strcmp(subject,"Physics")==0) tail->physics=new\_score;**

**if(strcmp(subject,"C")==0) tail->c=new\_score;**

**}**

**printf("Alter:\n%-15s%-20s%-10s%-10s%-10s%-10s\n","ID","Name","English","Math","Physics","C");**

**tail=head;**

**for(;tail!=NULL;tail=tail->next){**

**printf("%-15s%-20s%-10.2f%-10.2f%-10.2f%-10.2f\n",tail->num,tail->name,tail->english,tail->math,tail->physics,tail->c);**

**}**

**printf("\nSumAndAvg:\n");**

**tail=head;**

**for(;tail!=NULL;tail=tail->next){**

**tail->sum=tail->english+tail->math+tail->physics+tail->c;**

**tail->avg=tail->sum/4.0;**

**}**

**printf("%-15s%-20s%-10s%-10s\n","ID","Name","SUM","AVG");**

**tail=head;**

**for(;tail!=NULL;tail=tail->next){**

**printf("%-15s%-20s%-10.2f%-10.2f\n",tail->num,tail->name,tail->sum,tail->avg);**

**}**

**printf("\nSort:\n");**

**tail=head;**

**struct Student \*temp;**

**struct Student \*temp1=(struct Student \*)malloc(sizeof(struct Student));**

**for(;tail!=NULL;tail=tail->next)**

**for(temp=tail->next;temp!=NULL;temp=temp->next){**

**if(temp->avg<tail->avg){**

**Studentcopy(temp,temp1);**

**Studentcopy(tail,temp);**

**Studentcopy(temp1,tail);**

**}**

**}**

**printf("%-15s%-20s%-10s\n","ID","Name","AVG");**

**tail=head;**

**for(;tail!=NULL;tail=tail->next){**

**printf("%-15s%-20s%-10.2f\n",tail->num,tail->name,tail->avg);**

**}**

**printf("\n");**

**struct Student \*temp3=head;**

**struct Student \*temp2=temp3->next;**

**while(temp2!=NULL){**

**free(temp3);**

**temp3=temp2;**

**temp2=temp2->next;**

**}**

**free(temp3);**

**return 0;**

**}**

3）测试

（a） 测试数据：5

U20140101 ZhangChuanChao 85 86 87 88

U20140126 MaiDouDou 99 99 99 99

U20140158 XiaoDouDou 56 85 89 59

U20140312 DaoDaoDog 84 89 65 100

U20140359 XiDaDa 88.8 88.8 88.8 88.8

3

U20140101 Math 95.6

U20140359 C 100

U20140359 English 100

（b） 对应测试运行结果如图所示。

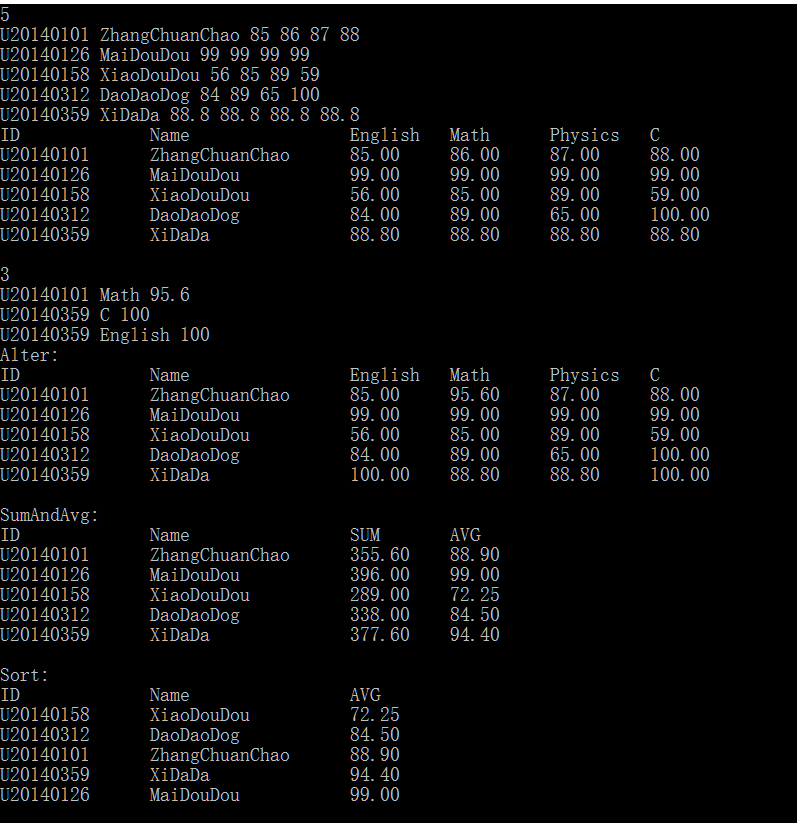


图7-2 编程题2的测试用例一的运行结果

## 7.3 实验小结

通过本次实验，我掌握了结构和联合的使用方法，尤其对于链表这种数据结构有了新的体会，也通过各种表达式的值进一步理解了指针的概念，锻炼了我的编程能力。

# 8 文件实验

## 8.1 实验目的

(1)熟悉文本文件和二进制文件在磁盘中的存储方式。

（2）熟练掌握流式文件的读写方法。

## 8.2 实验内容

**8.2.1 文件类型的程序验证**

设有程序：

#include<stdio.h>

int main(void){

short a=0x253f,b=0x7b7d;

char ch;

FILE \*fp1,\*fp2;

fp1=fopen("d:\\abc1.bin","wb+");

fp2=fopen("d:\\abc2.txt","w+");

fwrite(&a,sizeof(short),1,fp1);

fwrite(&b,sizeof(short),1,fp1);

fprintf(fp2,"%hx%hx",a,b);

rewind(fp1);rewind(fp2);

while((ch=fgetc(fp1))!=EOF)

putchar(ch);

putchar('\n');

while((ch=fgetc(fp2))!=EOF)

putchar(ch);

putchar('\n');

fclose(fp1);

fclose(fp2);

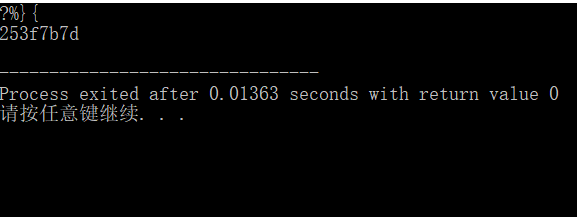
return 0;

}

1. 请思考程序的输出结果，然后通过上机运行来加以验证。

**输出为：**？%}{

235f7b7d



1. 将两处sizeof(short)均改成sizeof(char)后，结果有什么不同？为什么？

**结果变为：？} 因为修改前按short类型写入4个字节，修改后只写入两个字节**

1. 将fprintf(fp2,"%hx%hx",a,b);改成fprintf(fp2,"%d%d",a,b);后结果有什么不同。**结果将a，b转换成了10进制输出。**

**8.2.2 源程序修改替换**

（1）下面源程序中存在什么样的逻辑错误（先观察执行结果）？请对程序进行修改、调试，使之能够正确完成指定任务。

**源程序：**

#include<stdio.h>

#include<stdlib.h>

int main(int argc,char \*argv[]){

char ch;

FILE \*fp;

if(argc!=2){

printf("Arguments error!\n");

exit(-1);

}

if((fp=fopen(argv[1],"r"))==NULL){

printf("Cant't open %s file!\n",argv[1]);

exit(-1);

}

while(ch=fgetc(fp)!=EOF)

putchar(ch);

fclose(fp);

return 0;

}

**解答：在while中缺少括号，造成逻辑混乱。**

（2）用输入输出重定向freopen改写上述源程序中的main函数。

**解答：**

**#include<stdio.h>**

**#include<stdlib.h>**

**int main(int argc,char \*argv[]){**

**char ch;**

**FILE \*fp;**

**if(argc!=2){**

**printf("Arguments error!\n");**

**exit(-1);**

**}**

**if((fp=freopen(argv[1],"r",stdin))==NULL){**

**printf("Cant't open %s file!\n",argv[1]);**

**exit(-1);**

**}**

**while((ch=fgetc(fp))!=EOF)**

**putchar(ch);**

**fclose(fp);**

**return 0;**

**}**

**8.2.3 程序设计**

**（1）**编写一个程序，实现以下功能：从键盘输入一行英文句子，将每个单词的首字母换成大写字母，然后输出到一个磁盘文件“test”中保存。

**解答：**

1. **解题思路：**

读入字符串，对第一个字符特判，其余紧接着空格的字符调整大小写，利用freopen（）重定向函数将输出转换到特定文件中。

2）**源程序**

**//#include<iostream>**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<string.h>**

**//#include<algorithm>**

**//using namespace std;**

**char s[100];**

**int main(){**

**freopen("test.txt","w",stdout);//输出重定向**

**fgets(s,100,stdin);**

**int space=0;**

**if(s[0]>='a'&&s[0]<='z') s[0]-=32;**

**for(int i=1;s[i]!='\n';i++){**

**if(s[i]==' ') {**

**space++;**

**continue;**

**}**

**if(space>0){**

**if(s[i]>='a'&&s[i]<='z')s[i]-=32;**

**space=0;**

**}**

**}**

**printf("%s",s);**

**fclose(stdout);**

**return 0;**

**}**

3）测试

（a） 测试数据：

i have a blue pen!

（b） 对应测试数据的运行结果截图

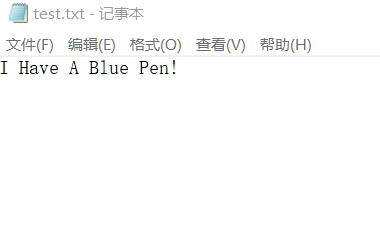


图8-1 编程题1的测试用例一的运行结果

## 8.3 实验小结

通过这次试验，我学会了进行文件输入输出的多种方式，为将来进行大型项目的开发打好了基础。加深了我对于二进制文件和普通文件的理解。

# 9.参考文献

《C语言实验与课程设计》------李开、卢萍、曹计昌