**华中科技大学**

C**语言实验与课程设计**

**实验报告**

**计算机科学与技术学院**

**指导老师：**

**目录**

**1 表达式和标准输入输出实验 ········ 3**

**2 流程控制实验 ·············· 11**

**3 函数程序与结构实验 ··········· 20**

**4 编译预处理实验 ············· 28**

**5 数组实验 ················ 35**

**6 指针实验 ················ 42**

**7 结构与联合实验 ············· 52**

**8 文件实验 ················ 61**

**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 实验内容及结果**

**1.2.1.** 源程序改错

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。要求参照2.1.3和2.1.4的步骤对下面程序进行调试修改，使之能够正确完成指定任务。

**程序代码：**

#include<stdio.h>

#define PI 3.14159;

voidmain( void )

{

int f ;

short p, k ;

double c , r , s ;

/\* for task 1 \*/

printf(“Input Fahrenheit:” ) ;

scanf(“%d”, f ) ;

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

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

/\* for task 2 \*/

printf("input the radius r:");

scanf("%f", &r);

s = PI \* r \* r;

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

/\* for task 3 \*/

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

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

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

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

}

【分析及改正】本程序共存在9处错误，分析如下：

（1）#define PI 3.14159；

错误原因：宏定义在预处理时会将后续程序中的标示符展开为对应全部字符串，而PI不需要分号。

改正：去掉其后分号

（2）voidmain(void) {

错误原因：函数返回值与函数名之间应加空格，C90标准建议main函数应返回一个int作为状态值

改正：int main(void) {

（3）short p,k;

错误原因：后续用到了newint变量但并未声明

改正：short p,k,newint;

（4）scanf(“%d”, f );

错误原因：应该把要读入的变量的地址传递给scanf才能正确写入

改正：scanf("%d",&f);

（5）c = 5/9\*(f-32) ;

错误原因：5和9是整型常数，相除会截断小数部分，应该将其中任意一个输表示为或转换为浮点数类型

改正：c = 5.0/9\*(f-32);

（6）scanf("%f", &r);

错误原因：double类型对应的占位符是%lf

改正：scanf("%lf",&r);

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

错误原因：要输出s的值应该将s作为参数传递给printf，不应传递其地址

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

（8）scanf("%x %x", &k, &p );

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

错误原因：k,p为short类型，对应十六进制占位符应为%hx

改正：scanf("%hx %hx",&k,&p);

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

（9）newint = (p&0xff00)|(k&0xff00)<<8;

错误原因：要求将k的高字节作为结果的低字节，应该将k的高字节提取出后右移至低字节

改正：newint=(p&0xff00)|(((k&0xff00)>>8)&0x00ff);

**1.2.2 源程序修改替换**

下面的程序利用常用的中间变量法实现两数交换，请改用不使用第3个变量的方法实现。该程序中t是中间变量，要求将定义语句中的t删除，修改下划线处的语句，使之实现两数对调的操作。

**程序代码：**

#include<stdio.h>

void main( )

{

int a, b, t;

printf(“Input two integers:”);

scanf(“%d %d”,&a,&b);

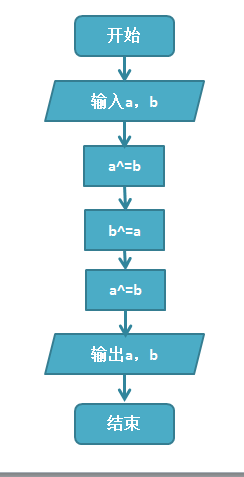
t=a ；a=b；b=t；

prinf(“\na=%d,b=%d”,a,b);

}

【分析】将a,b中的任何一个看做原码，另一个当做密钥进行加密，再将加密过的内容与密钥做加密运算即得到原码。由此使用异或运算进行交换。

**流程图如下：**



【程序】

#include <stdio.h>

#include <stdlib.h>

int main(){

int a,b;

printf("input two integers:");

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

a ^= b;

b ^= a;

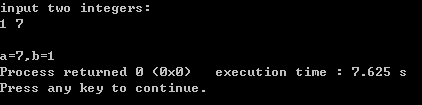
a ^= b;

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

return 0;

}

【测试】如果输入a= 1，b=7，应该输出：a=7，b=1，程序运行结果如下，程序正确。

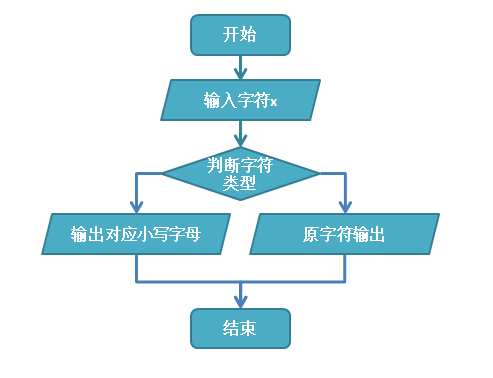


**1.2.3 程序设计**

**1、**编写一个程序，输入字符ｃ，如果ｃ是大写字母，则将ｃ转换成对应的小写，否则ｃ的值不变，最后输出ｃ。

【分析】因为同一个字母的大小写字母的ASCII码值相差32，若ch为大写字母，则ch+32为对应的小写字母。

**流程图如下：**



【程序】

#include <stdio.h>

int main()

{

printf("Input your char x:\n");

char x;

scanf("%c",&x);

if(x>='a'&&x<='z')

{

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

}

else if(x>='A'&&x<='Z')

{

printf("%c\n",x-'0'+32+'0');

}

else{

printf("Please input the right char.");

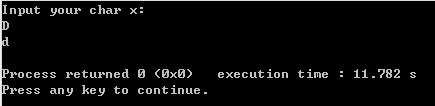
}

return 0;

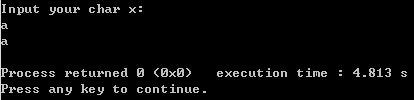
}

【测试】

（1）测试大写变小写。若输入D，应输出d，程序运行结果如下，程序正确。



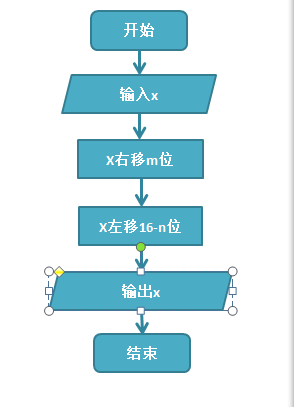
（2）测试其他字符。若输入a，应输出a，程序运行结果如下，程序正确。



2、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。

【分析】将x首先右移m位，再左移(16-n)位，则末尾会补0，即可将第m位开始向左的n位且向最高位靠齐

流程图如下：



【程序】

#include <stdio.h>

int main(){

unsigned short x,m,n;

puts("Please input x,m,n:");

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

x>>=m;

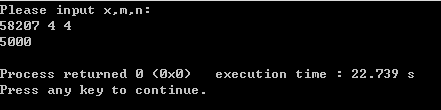
x<<=(16 - n);

printf("%hx\n",x);

return 0;

}

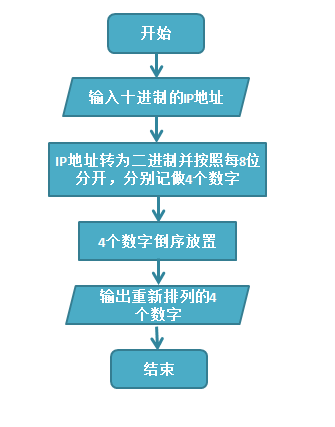
【测试】若 x=58207，m=4，n=4，因为(58207)10 = (1110 0011 0101 1111)2，处理后为(101 0000 0000 0000)2= (5000)16 ，程序应该输出十六进制数：5000，运行结果如下，程序正确。



3、IP地址通常是4个用句点分隔的小整数（即点分十进制），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。例如，整形676879571二进制表示就是：00101000 01011000 01011100 11010011，按照8位一组可表示为：40 88 92 211，由于CPU处理数据的差异，它的顺序是颠倒的，所有最终格式为211.92.88.40。

【分析】分别利用逻辑尺取出读入的长整形数的四部分，从最低位起每8位（1个字节）为一个部分，然后按照正常顺序输出其对应的十进制IP地址。

程序框图如下：



【程序】

#include <stdio.h>

int main(){

unsigned long IP;

unsigned short first, second, third, fourth;

puts("Input the IP in unsigned long int form:");

scanf("%lu",&IP);

first = (IP&0xFF000000)>>24;

second = (IP&0xFF0000)>>16;

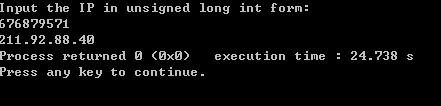
third = (IP&0xFF00)>>8;

fourth =IP&0xFF;

printf("%hu.%hu.%hu.%hu",fourth,third,second,first);

}

【测试】若输入676879571，应输出 211.92.88.40。运行结果如下，表明程序正确。



**1.3 实验体会**

在做改错题时，先浏览一遍，找出明显的错误，然后将初步修改后的代码输入编译器并编译，再根据编译器的警告与错误提示修改程序至完全正确，这个过程很需要细心和耐心，这个过程能够很好的帮助我再今后不犯类似错误。

源程序修改与替换题锻炼我们举一反三的能力，我们在平时写程序时就应该思考对于一个问题有没有多钟解决方案，不使用中间变量交换两个整数是很常见的问题，我利用一个数异或自身等于0，0异或任意位保留原位的原理来交换。

程序设计题提高了我们对所学知识的熟练度和理解程度，只有实践才能学会C语言，在亲手编写代码的过程中发现问题，理解概念。

**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）熟练运用for、while、do-while语句来编写程序。

（3）练习转移语句和标号语句的使用。

（4）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。

2.2实验内容及要求

2.2.1. 源程序改错

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源程序进行调试修改，使之能够正确完成指定任务，例如，8!=40320。

**程序代码：**

#include<stdio.h>

int main()

{

int i,n,s=1;

printf("Please enter n:");

scanf("%d",n);

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

s=s\*i;

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

return 0;

}

【分析及改正】本程序共存在3处错误，分析如下：

（1）int s=1;

错误原因：在计算阶乘的时候，由于阶乘值会上升很快，所以不建议使用int类型，可以改为long或者double型。

改正：改为long型变量。

（2）scanf(“%d”,n);

错误原因：scanf函数获取输入值时，引用的是一个变量地址，而不是变量本身。

改正：scanf(“%d”,&n);

（3）} return 0;

错误原因：return返回值应该在main函数内部

改正：return 0;}

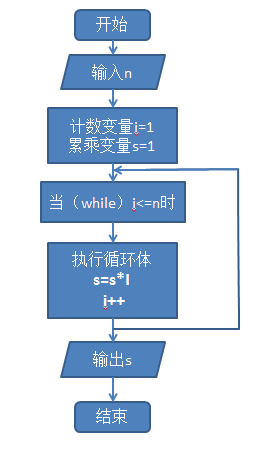
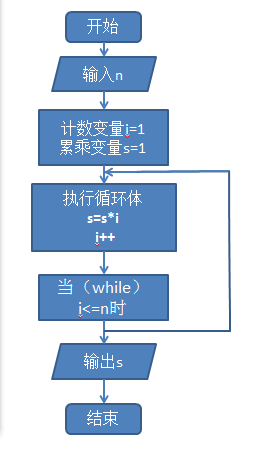
2.2.2 源程序修改替换

（1）修改上述计算s=n!的源程序，要求分别用while和do-while语句替换for语句。

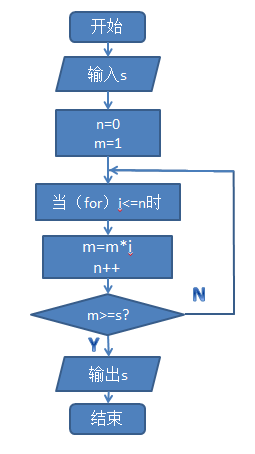
（2）修改上述计算s=n!的源程序，要求输入改为“整数S”，输出改为“满足n! >=s的最小整数n”。例如，输入整数40310，输出结果为n=8.

【分析】（1）相同循环内容利用不同循环结构实现，需要注意不同语句的格式及根据循环方式对循环体做些许变动。流程图如下：

**while型 do-while型**

（2）同样计算n!并将n!赋值给不是s的值（比如说m），然后令m和s比较大小，找到符合条件的第一个m值，并返回n值。流程图如下：



【程序】（1）**while型**

#include <stdio.h>

int main()

{

int i=1,n,s=1;

printf("Please enter n:");

scanf("%d",&n);

while(i<=n){

s=s\*i;

i++;}

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

return 0;

}

**do-while型**

#include <stdio.h>

int main()

{

int i=1,n,s=1;

printf("Please enter n:");

scanf("%d",&n);

do{

s=s\*i;

i++;

}

while(i<=n);

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

return 0;

}

（2）

#include <stdio.h>

int main()

{

int i,n=0,s,m=1;

printf("Please enter s:");

scanf("%d",&s);

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

m=m\*i;

n++;

if(m>=s){

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

break;

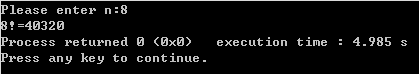
}

}

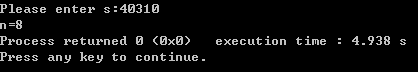
return 0;

}

【测试】（1）如果输入8，应该输出：40320，程序运行结果如下，程序正确。



（2）如果输入40310，应该输出：8，程序运行结果如下，程序正确。



2.2.3 程序设计

编写并上机调试运行能实现下列功能的程序：

（1）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%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语句呵switch语句来实现。

（2）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。

（3）编写一个程序，打印如下的杨辉三角形。

1

1. 1

1 2 1

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

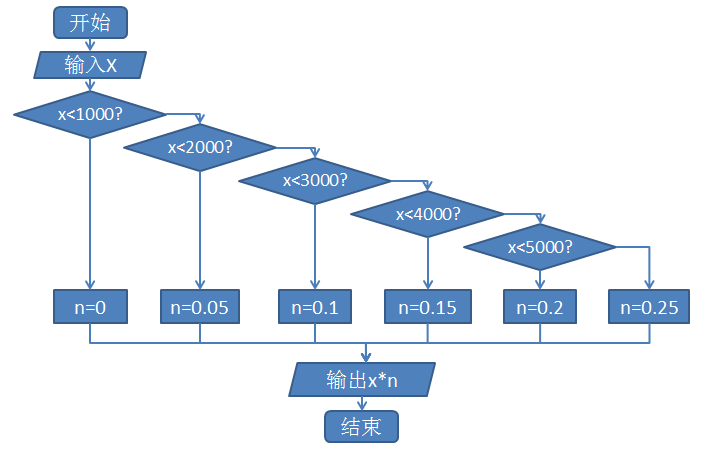
1 9 36 84 126 126 84 36 9 1

（4）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。

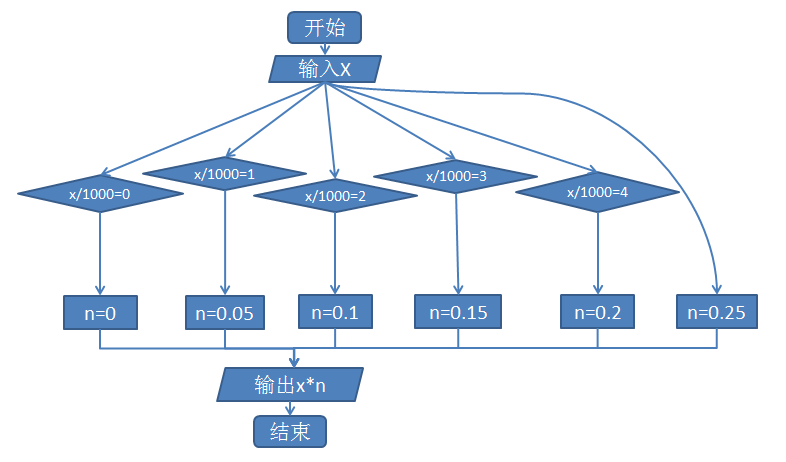
【分析】

（1）数学中的分段函数问题，可以用if-else-if结构进行控制。在应用switch语句的时候，结合分段区间的特点设置合适的间断点。流程图分别如下：

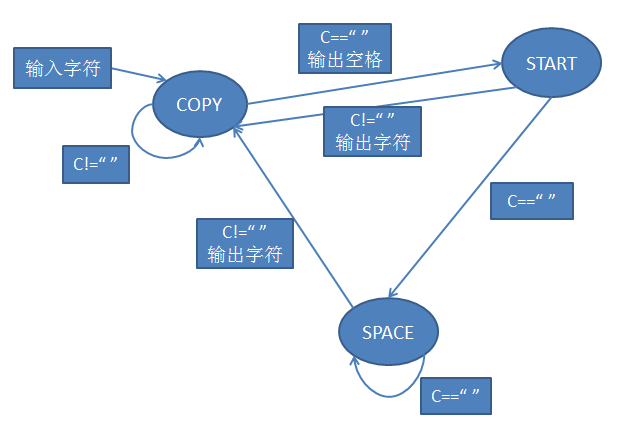
if型



switch型

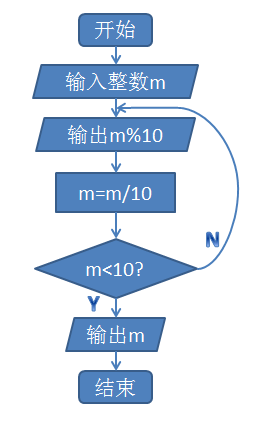


（2）使用状态机通过判断空格的个数选择不同状态，从而实现消去空格的作用。流程图如下：



（3）根据所给公式利用循环实现杨辉三角数的计算，注意题目中特别提到的输出格式，将每个数字设置为四个字符位，并循环输出空格符，实现金字塔效果。

（4）多次除10取余，每次都分别输出余数，然后除以10并赋给原数重复计算，流程图如下：



【程序】

（1）**if型**

#include <stdio.h>

float tax(float x);

int main()

{

float x;

printf("Please input your salary:");

scanf("%f",&x);

printf("You should pay %f yuan.\n",tax(abs(x)));

return 0;

}

float tax(float x){

float n;

if(x<1000){n=0;}

else if(x<2000){n=0.05;}

else if(x<3000){n=0.1;}

else if(x<4000){n=0.15;}

else if(x<5000){n=0.2;}

else{n=0.25;}

return n\*x;

}

**switch型**

#include <stdio.h>

float tax(float x);

int main()

{

float x;

printf("Please input your salary:");

scanf("%f",&x);

printf("You should pay %f yuan.\n",tax(abs(x)));

return 0;

}

float tax(float x){

float n;

switch((int)(x/1000)){

case 0:n=0;break;

case 1:n=0.05;break;

case 2:n=0.1;break;

case 3:n=0.15;break;

case 4:n=0.2;break;

default:n=0.25;

}

return n\*x;

}

（2）

#include <stdio.h>

enum {COPY,START,SPACE};

int main()

{

char c;

int state=COPY;

printf("Please input a char:\n");

while((c=getchar())!='\n'){

switch(state){

case COPY:

if(c==' ')state=START;

else putchar(c);

break;

case START:

putchar(' ');

if(c==' ')state=SPACE;

else{state=COPY;putchar(c);}

break;

case SPACE:

if(c!=' '){

state=COPY;

putchar(c);

}

else state=SPACE;

break;

}

}

return 0;

}

（3）

#include <stdio.h>

#define n 10

int main(void)

{

int i,j,k,c,m;

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

{

for(m=0;m<=n-i;m++){

printf(" ");

}

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

{

if(j==0)c=1;

else

c=c\*(i+1-j)/j;

printf("%4d",c);

}

printf("\n");

}

return 0;

}

（4）

#include <stdio.h>

int main()

{

long n,m;

printf("Please input an integer n(n>0):");

scanf("%ld",&n);

printf("\nThe new integer is:");

while(n>=10){

m=n%10;

printf("%ld",m);

n/=10;

}

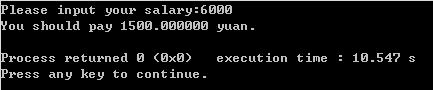
printf("%ld",n);

return 0;

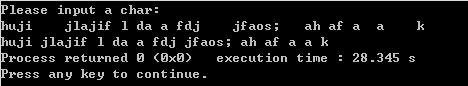
}

【测试】

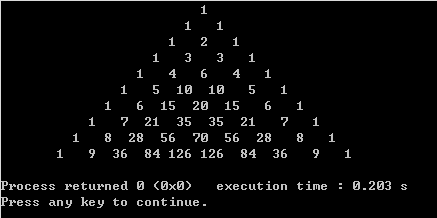
（1）



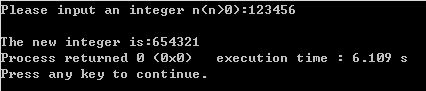
（2）



（3）



（4）



2.3 实验体会

源程序改错中，依赖于平时良好的输入习惯，在输入过程中已经可以检查出大多数语法错误，或者拼写错误。接下来再检查一下逻辑是否有误，无误则进行多组数据的编译检验，确认最终无误。

源程序修改与替换题锻炼我们举一反三触类旁通的能力，在平时写程序时就应该思考对于一个问题有没有多钟解决方案，使用不同的逻辑结构多种尝试寻求最优解。

程序设计题则是自主的设计实践，在练习中获得进一步提升，增强对于所学知识的熟练度和理解程度，锻炼能力。

**3函数与程序结构实验**

2.1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。

（4）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。

2.2实验内容及要求

2.2.1. 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源程序进行调试修改，使之能够正确完成指定任务。

**程序代码：**

#include<stdio.h>

int main(void)

{

int k;

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

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

}

return 0;

longsum\_fac(int n)

{

long s = 0;

int i;

long fac;

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

s += fac;

return s;

}

【分析及改正】本程序共存在3处错误，分析如下：

（1）} return 0;

错误原因：return返回值应该在main函数内部

改正：return 0;}

（2）long fac;

错误原因：fac未赋予初始值，会导致接下来计算错误。

改正：long fac=1;

（3）s += fac;

错误原因：语句s += fac;不在函数for内部，逻辑出错。

改正：for(i=1;i<=n;i++){

fac\*=i;

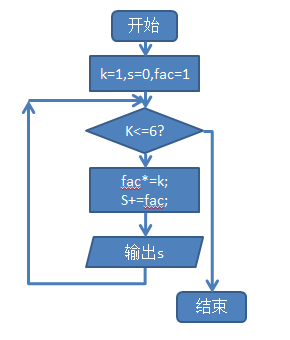
s+=fac; }

2.2.2 源程序修改替换

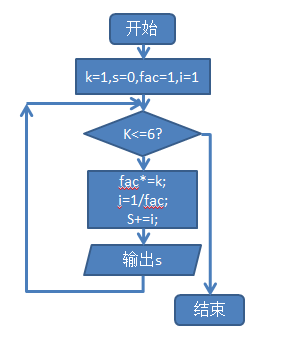
（1）修改上术源程序中的sum\_fac函数，使其计算量最小。

（2）修改上述源程序中的sum\_fac函数，计算s =1+1/(2!)+1/(3!)+……+1/(n!)

【分析】（1）利用static变量类型对循环进行化简，保留上一个值，避免多次重复循环。流程图如下：



（2）相比较于源程序，变动之处在于求的是阶乘的倒数和，则在循环中多一步求倒数的过程，并对倒数求和。流程图如下：



【程序】（1）

#include <stdio.h>

long sum\_fac();

int main()

{

int k;

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

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

return 0;

}

static long s=0,fac=1;

long sum\_fac(int n)

{

fac\*=n;

s+=fac;

return s;

}

（2）

#include <stdio.h>

float sum\_fac();

int main()

{

int k;

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

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

return 0;

}

float sum\_fac(int n)

{

long fac=1;

int i;

float x,s=0;

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

fac\*=i;

x=1.0/fac;

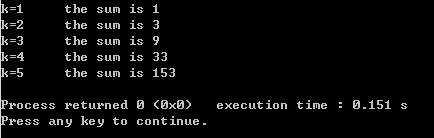
s+=x;

}

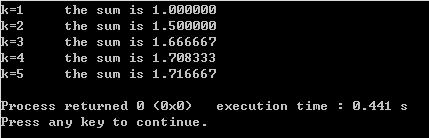
return s;

}

【测试】（1）程序运行结果如下，程序正确。



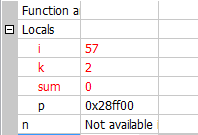
（2）程序运行结果如下，程序正确。



2.2.3跟踪调试

下面是计算fabonacci数列前n项和的源程序，现要求单步执行该程序，并观察p，i，sum，n的值，即：

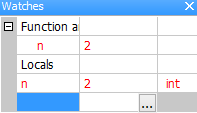
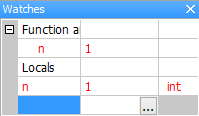
（1）刚执行完“scanf(”%d”,&k);”语句时，p,i值是多少？



（2）从fabonacci函数返回后，光条停留在哪个语句上？

C:\Users\q\AppData\Roaming\Tencent\Users\479101295\QQ\WinTemp\RichOle\WJW2@)%(H5V}O]Y%KFQBPDN.png

（3）进入fabonacci函数时，watches窗口显示的是什么？

等图像。

（4）当i=3时，从调用fabonacci函数到返回，n值如何变化？

n=1到n不存在；n=1到n=2到n不存在；n=1到n=2到n=3到n不存在。

【源程序】

#include <stdio.h>

long fabonacci(int);

int main()

{

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);

}

return 0;

}

long fabonacci(int n){

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

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

}

2.2.4 程序设计

编写并上机调试运行能实现以下功能的程序：

（1）编写一个程序，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，观察其递归过程。

（2）编写一个程序，验证哥德巴赫猜想：一个大于等于4 的偶数都是两个素数之和。

（3） 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。如， 如果BEGIN为10，END为20，则程序的输出应为：

GOLDBACH’SCONJECTURE:

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

10=3+7

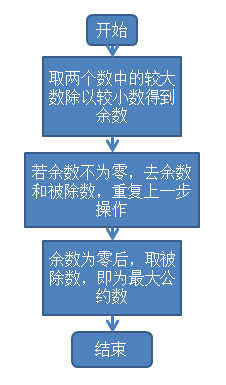
12=5+7

……

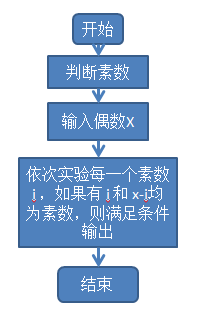
20=3+17

【分析】

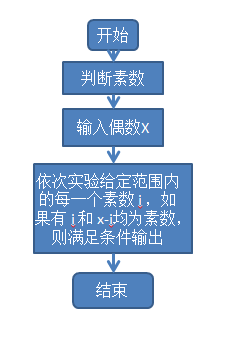
（1）用辗转相除法求最大公约数，程序框图如下：



（2）首先判断一定范围内的素数（比如100以内），然后判断一个偶数能否表示为两个素数之和。程序框图如下：



（3）和第二题类似，只不过加上一个范围控制，设置循环处于最大值和最小值之间。程序框图如下：



【程序】

（1）

#include <stdio.h>

int GCD(int m,int n);

int main()

{

int m,n,r;

printf("Please input two integers:\n");

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

printf("The greatest common divisor is %d",GCD(m,n));

return 0;

}

int GCD(m,n){

if(m%n==0&&m>=n)return n;

else if(m%n==0&&m<n)return m;

else{

int r;

r=m%n;

m=n;

n=r;

return GCD(m,n);

}

}

（2）

#include <stdio.h>

#define N 100

static int a[N];

int prime();

int main()

{

int x,i;

printf("Please input an even x(x>=4):\n");

scanf("%d",&x);

if(x%2!=0||x<4)printf("ERROR!");

prime();

for(i=2;i<=N&&i<x/2;i++){

if(a[i])

if(a[x-i]){printf("%d=%d+%d\n",x,i,x-i);}

}

return 0;

}

int prime(){

int i,j;

for(a[0]=a[1]=0,i=2;i<N;i++)a[i]=1;

for(i=2;i<N/2;i++)

if(a[i]){

for(j=i\*i;j<N;j+=i)a[j]=0;

}

return 0;

}

（3）

#include <stdio.h>

#define N 100

#define BEGIN 10

#define END 20

static int a[N];

int prime();

int main()

{

int x,i,j;

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

prime();

for(i=BEGIN;i<=END;i++){

for(j=2;j<i/2;j++){

if(a[j])

if(a[i-j]){printf("%d=%d+%d\n",i,j,i-j);}

}

}

return 0;

}

int prime(){

int i,j;

for(a[0]=a[1]=0,i=2;i<N;i++)a[i]=1;

for(i=2;i<N/2;i++)

if(a[i]){

for(j=i\*i;j<N;j+=i)a[j]=0;

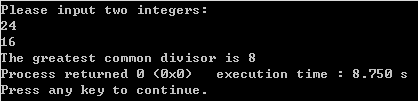
}

return 0;

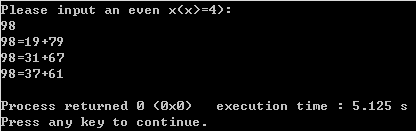
}

【测试】

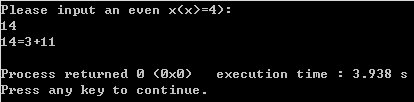
（1）输入24和16求最大公约数，最大公约数为8，结果正确：



（2）验证哥德巴赫猜想， 输入98，返回素数和，结果正确：



（3）验证一定范围内的哥德巴赫猜想，输入14，返回素数和，结果正确：



2.3 实验体会

源程序改错中，依赖于平时良好的输入习惯，在输入过程中已经可以检查出大多数语法错误，或者拼写错误。接下来再检查一下逻辑是否有误，无误则进行多组数据的编译检验，确认最终无误。

源程序修改与替换题锻炼我们优化算法的能力，在平时写程序时就应该思考对于一个问题有没有最优解决方案，使用不同的语句在空间和时间中寻求平衡点，尽可能的控制好占用的时间空间范围。

程序设计题则是自主的设计实践，在练习中获得进一步提升，增强对于所学知识的熟练度和理解程度，锻炼编写代码的能力以及并通过问题的进一步引申锻炼思维的灵活性。

**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和交换两数的源程序。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源程序进行调试修改，使之能够正确完成计算平方差和交换两数的任务。

**程序代码：**

#include<stdio.h>

#define SUM a+b

#define DIF a–b

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

int main(void)

{

int b, t;

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);

return 0;

}

【分析及改正】本程序共存在3处错误，分析如下：

（1）#define SUM a+b

#define DIF a–b

错误原因：在宏定义中，在引用时直接将表达式带入之后的表达式应用，由于不了解在之后的应用中采用的运算符，所以优先级无法确定，因此需要在宏定义中对变量即表达式中加上括号。

改正：#define SUM ((a)+(b))

#define DIF ((a)–(b))

（2）#define SWAP(a,b) a=b,b=a

错误原因：带参数的宏定义逻辑错误，无法有效实现两个数的交换。

改正：#define SWAP(a,b,t) t=b,b=a,a=t

（3）int b, t;

错误原因：未完整声明所有参数，导致部分参数不识别。

改正：int a,b,t;

（4）scanf("%d, %d", &a, &b);

错误原因：scanf函数中类型声明中，两个%d符号中间不加符号。

改正：scanf("%d%d", &a, &b);

（5）SWAP(a,b);

错误原因：与宏定义相对应的更改为三个参数。

改正：SWAP(a,b,t);

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、计算两数之和的源程序。在这个源程序中存在若干语法和逻辑错误，要求：

（1）对这个例子源程序进行调试修改，使之能够正确完成指定任务。

（2）用带参数的宏替换函数max，实现求最大数的功能。

**程序代码：**

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));

return 0;

}

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;

}

【分析】（1）修改如第一题，出现的问题有：scanf函数的参数变量类型间的格式问题与函数声明问题。

（2）将函数max用带参数的宏定义表示，利用语句“？：”表示if选择语句。

【程序】（1）

#include <stdio.h>

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

float sum(float x,float y);

int main()

{

int a,b,c;

float d,e;

printf("Enter three integers:\n");

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

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

printf("Enter two float point numbers:\n");

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 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;

}

（2）

#include <stdio.h>

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

float sum(float x,float y);

int main()

{

int a,b,c;

float d,e;

printf("Enter three integers:\n");

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

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

printf("Enter two float point numbers:\n");

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

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

return 0;

}

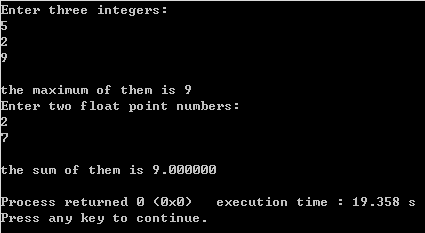
float sum(float x,float y)

{

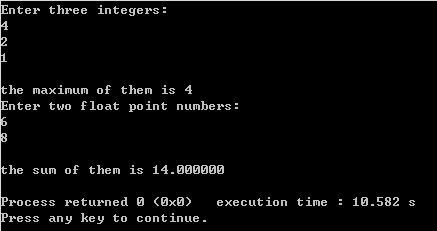
return x+y;

}

【测试】（1）程序运行结果如下，程序正确。



（2）程序运行结果如下，程序正确。

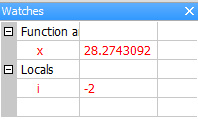
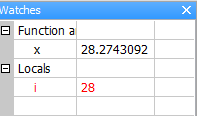


4.2.3跟踪调试

下面程序的功能是利用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。修改过的源程序见下面部分。

（2）单步执行。进入函数integer\_fraction时，watches窗口中为何值？在返回main时， watch窗口中i为何值？如图：

x的值  i的值 

（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。

题目程序代码：

#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);

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

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

#endif

return 0;

}

int integer\_fraction(float x)

{

int i = x;

return i;

}

【修改后的源程序】

#include <stdio.h>

#include <stdlib.h>

#include <assert.h>

#define R

int integer\_fraction(float x);

int main()

{

float r,s;

int s\_integer=0;

printf("Input a number:\n");

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 // R

return 0;

}

int integer\_fraction(float x)

{

int i=x;

return i;

}

4.2.4 程序设计

编写并上机调试运行能实现以下功能的程序：

（1）已知三角形的面积是area=根号下[s(s–a)(s–b)(s–c)]，其中s= (a+b+c)/2，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，试编写一程序，用带参数的宏来计算三角形的面积。

（2）使用条件编译方法编写一程序，其功能要求是，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，其他字符不变。用#define命令控制是否变换字母的大小写。

【分析】

（1）根据公式用宏定义表示s和area，并分别在函数中引用。

（2）宏定义变量CHANGE，通过改变CHANGE的值，利用if-else语句，控制大小写变换或者原文输出。

【程序】

（1）

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define s(a,b,c) 1.0/2\*((a)+(b)+(c))

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

int exam(int a,int b,int c);

int main()

{

double a,b,c;

printf("Please input three numbers:\n");

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

if(abs(a-b)<c&&abs(b-c)<a&&abs(a-c)<b)

printf("s=%lf area=%lf",s(a,b,c),area(s(a,b,c),a,b,c));

else printf("ERROR!");

return 0;

}

（2）

#include <stdio.h>

#include <stdlib.h>

#define CHANGE 1

int main()

{

char c;

printf("Please input the paragraph end with ctrl+z:\n");

while((c=getchar())!=EOF)

{

if(CHANGE)

{

if(c>='A'&&c<='Z')putchar(c+32);

else if(c>='a'&&c<='z')putchar(c-32);

else putchar(c);

}

else putchar(c);

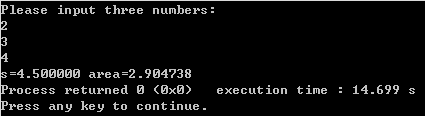
}

return 0;

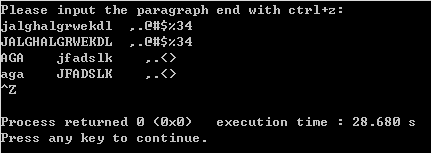
}

【测试】

（1）输入三边值为2，4，结果正确：

,

（2）CHANGE值为1时，结果正确：



4.3 实验体会

源程序改错中，依赖于平时良好的输入习惯，在输入过程中已经可以检查出大多数语法错误，或者拼写错误。接下来再检查一下逻辑是否有误，无误则进行多组数据的编译检验，确认最终无误。

源程序修改与替换题锻炼我们将函数功能转化为带参数的宏定义的能力，同时要注意宏定义的引用方式，时刻注意加上括号，防止在引用过程中由于优先级不同导致引用产生歧义，严谨杜绝副作用。

程序设计题则是自主的设计实践，根据题目要求自主设计宏定义，增强对于所学知识的熟练度和理解程度，锻炼编写代码的能力以及并通过问题的进一步引申锻炼思维的灵活性。

**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的实现算法。

（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。

**程序代码：**

#include<stdio.h>

int main(void){

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

void sort(int [],int);

int i;

sort(a[0],10);

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

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

printf("\n");

return 0;

}

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

int i,j,t;

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

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

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

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

}

【分析及改正】本程序共存在3处错误，分析如下：

（1）void sort(int [],int);

错误原因：函数声明位置错误，应该在main函数之前，声明以后才可以再main函数内部调用。

改正：void sort(int [],int);

int main()

（2）sort(a[0],10);

错误原因：调用sort函数时，第一个参数类型为指针型，所以引用的参数值应该是a，表示一个地址，即数组a的首地址。而不是a[0]，一个具体的数值。

改正：sort(a,10);

（3）t=b[j],b[j]=b[j+1],b[j+1]=t;

错误原因：在语句和语句之间使用分号。

改正：t=b[j];b[j]=b[j+1];b[j+1]=t;

5.2.2 源程序修改替换

（1）下面的源程序用于求解瑟夫问题：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，知道圈中只剩下一个人为止。请在源程序2中的下划线处填写合适的代码来完善该程序。

（2）上面的源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。如果采用做标记的方法，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，请采用做标记的方法修改（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。

**程序代码：**

#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++) /\*对圈中人安顺序编号1~M\*/

a[i]=i+1;

for(i=M, j=0; i>1; i--){/\*i表示圈中人的个数，初始为M个，剩一个时循环结束；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?\_\_\_\_\_\_\_\_：\_\_\_\_\_\_\_;/\*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\*/

\_\_\_\_\_\_=\_\_\_\_\_\_\_;

}

for(i=0;i<M-1;i++)/\*按次输出出圈人的编号\*/

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

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

return 0;

}

【分析】（1）按照题目要求及程序中的注释理解题意，将被提出成员序号装入新数组，并将原数组一次向前移动。重复循环直到最后一个。

（2）对数组进行标记，利用0、1值对数组进行真值判断，确定数组元素是否在数组中，从而省去清除数组的步骤。

【程序】（1）

#include <stdio.h>

#define M 10

#define N 3

int main()

{

int a[M],b[M];

int i,j,k;

for(i=0;i<M;i++)a[i]=i+1;

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

for(k=1;k<=N;k++)

if(++j>i-1)j=0;

b[M-i]=j?a[j-1]:a[i-1];

if(j)

for(k=--j;k<i;k++)

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

}

for(i=0;i<M-1;i++)printf("%6d",b[i]);

printf("%6d\n",a[0]);

return 0;

}

（2）

#include <stdio.h>

#define M 10

#define N 3

int main()

{

int a[M]={0},b[M];

int i,j,k;

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

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

if(++j>M-1)j=0;

if((j==0&&a[9]==1)||(j!=0&&a[j-1]==1))k=k-1;

}

b[M-i]=j?j:M;

a[j-1]=1;

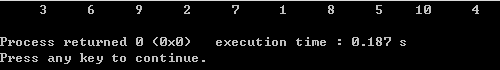
}

for(i=0;i<M;i++)printf("%6d",b[i]);

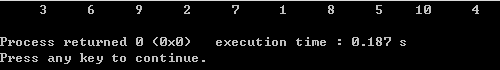
return 0;

}

【测试】（1）程序运行结果如下，程序正确。



（2）程序运行结果如下，程序正确。



5.2.3跟踪调试

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“}”所在行时，s、t分别为何值？

进入时C:\Users\q\AppData\Roaming\Tencent\Users\479101295\QQ\WinTemp\RichOle\B5F5YT%RI}O19BGLA~N8])1.png 结束时C:\Users\q\AppData\Roaming\Tencent\Users\479101295\QQ\WinTemp\RichOle\V~CSKE)BAH2)UM]ZF@[5YIB.png

（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。

题目程序代码：

#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';

}

【修改后的源程序】

#include <stdio.h>

#include <stdlib.h>

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

int main()

{

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])i++;

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

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

s[i]='\0'; }

5.2.4 程序设计

编写并上机调试运行能实现以下功能的程序：

（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）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。

（3）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，姓名同时作相应的调整，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”。

【分析】

（1）利用二位数组下标值的不同交换行列数组元素，实现转置数组的目的。

（2）由于十进制数在电脑中自动储存时也为二进制储存，所以不需要考虑转换的问题，只需要按照二进制由低位到高位输出，利用位运算就可以实现。

（3）二分法查找搜索，依次使用每一轮的最高位和最低位求出中间值并进行比较。

【程序】

（1）

#include <stdio.h>

#define M 3

#define N 4

void output(int m,int n,float a[][n])

{

int i,j;

for(i=0;i<m;i++){

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

printf("%10f",a[i][j]);

printf("\n"); }

}

int main()

{

int i,j;

float a[M][N],b[N][M];

printf("Please input an array item for 3\*4:\n");

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

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

scanf("%f",&a[i][j]);

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

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

b[j][i]=a[i][j];

printf("Array a is:\n");

output(M,N,a);

printf("\nArray b is:\n");

output(N,M,b);

return 0;

}

（2）

#include <stdio.h>

#define M 20

int main()

{

int i,a[M];

unsigned n;

printf("Please input an unsigned integer:\n");

scanf("%u",&n);

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

a[i]=n&0x0001;

n>>=1;

}

for(i=M-1;i>=0;i--)printf("%d",a[i]);

return 0;

}

（3）

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#define N 30

int main()

{

int i,j,temp;

char name[N][20],a[20]={'\0'},b;

int score[N];

for(i=0;i<N;i++){

printf("name:");

b=getchar();

gets(name[i]);

printf("score:");

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

}

for(i=0;i<N-1;i++){

for(j=0;j<N-1;j++){

if(score[j]<score[j+1]){

temp=score[j];

score[j]=score[j+1];

score[j+1]=temp;

strcpy(a,name[j]);

strcpy(name[j],name[j+1]);

strcpy(name[j+1],a); }

}

}

printf("\n\n");

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

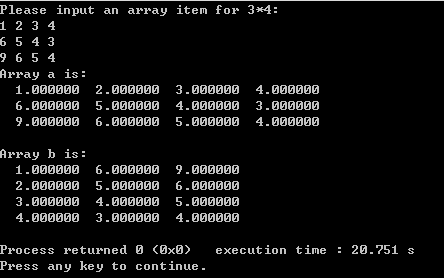
printf("name:%-15ssocre:%4d\n",name[i],score[i]);

return 0;

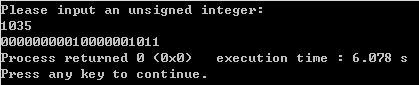
}

【测试】

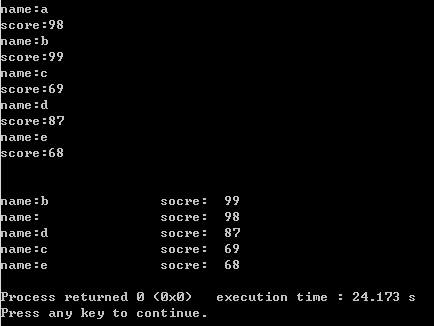
（1）结果正确：



（2）结果正确：



（3）结果正确：



5.3 实验体会

源程序改错中，依赖于平时良好的输入习惯，在输入过程中已经可以检查出大多数语法错误，或者拼写错误。接下来再检查一下逻辑是否有误，无误则进行多组数据的编译检验，确认最终无误。

源程序修改与替换题锻炼我们将充分利用数组的能力，同时要注意数组下标值是从0开始的，时刻注意它和现实中的计数的不同，防止在引用过程中由于对应不同导致产生逻辑错误，严谨杜绝副作用。

程序设计题则是自主的设计实践，根据题目要求自主设计数组，增强对于所学知识的熟练度和理解程度，锻炼编写代码的能力以及并通过问题的进一步引申锻炼思维的灵活性。

**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实验内容及要求

6.2.1. 源程序改错

下面的源程序中是否存在错误？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源程序进行调试修改，使之能够正确执行。

**程序代码：**

#include<stdio.h>

int main(void)

{

float \*p;

scanf("%f", p);

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

return 0;

}

【分析及改正】本程序错误为：指针在声明时未曾初始化就进行赋值操作，导致悬挂指针的存在，输入地址混乱，会使程序崩溃。

改正为：

#include <stdio.h>

int main()

{

float x,\*p=&x;

scanf("%f",p);

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

return 0;

}

6.2.2 源程序修改替换

（1）下面的源程序的功能是：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。请在程序中的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。

（2）请上机运行第（1）题程序，使之能按下面要求输出结果：

1 copy string.

2 connet string.

3 exit input a number(1-2) 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.

**程序代码：**

#include<stdio.h>

int main(void)

{

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:

gotodowm;

}

getchar();

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

i = 0;

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

i = 0;

result = (a, b);

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

dowm:

;

return 0;

}

【分析】（1）通过下面关于result的的操作可以看出，前面是用两个函数来控制选择哪个菜单来执行，因此p的作用就是等于不同的函数来起到开关的作用，所以前面需要声明p的存在，由此填出三个空上的内容。第一个是char \*(\*p)(char \*a,char \*b);第二个是gets(a)，第三个是gets(b)，第四个是p。

（2）由（1）可得正确的源程序，按照要求输入则可得到输出结果。

【程序】

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

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

gets(a);

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

i=0;

gets(b);

result=p(a,b);

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

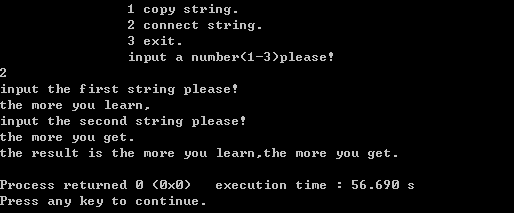
down:

;

return 0;

}

【测试】程序运行结果如下，程序正确。



6.2.3跟踪调试

请按下面的要求对所给的源程序进行操作，并回答问题和排除错误。

（1）单步执行源程序。进入strcpy时，watches窗口中s为何值？返回main时，watches 窗口中s为何值？

进入函数时C:\Users\q\AppData\Roaming\Tencent\Users\479101295\QQ\WinTemp\RichOle\S4TH]H1`AW~}%[`SI6~JD]C.png

返回main时，无s的值。

（2）排除源程序中的错误，使程序输出结果为：there is a bote on the lake.

题目程序代码：

#include<stdio.h>

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

int main(void)

{

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

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

return 0;

}

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

{

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

;

return (s);

}

【修改后的源程序】

#include <stdio.h>

#include <stdlib.h>

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

int main()

{

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

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

return 0;

}

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

{

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

;

return(s);

}

6.2.4 程序设计

编写并上机调试运行能实现以下功能的程序：

（1）已知一个长整型变量占4个字节，其中每个字节又分成高4位和低4位。试编写一 个程序，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记性显示。

（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。试编 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。

（3）编写一个程序，输入n个整数，排序后输出。排序的原则由命令行可选参数-的-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。

（4）设某个班有N个学生，每个学生修了M门课程。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M门课程的成绩，并且都存放到相应的数组中。试编写下列函数：

① 计算每个学生各门课程平均成绩。

② 计算全班没门课程的平均成绩。

③ 分别统计低于全班各门课程平均成绩的人数。

④ 分别统计全班各门课程不及格的人数和90分以上的人数。

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，不得使用下下标操作）

【程序】

（1）

#include <stdio.h>

void litoc(long int);

int main()

{

long int a;

printf("please input a long integer:\n");

scanf("%ld",&a);

litoc(a);

return 0;

}

void litoc(long int a)

{

char \*p,low,high;

p=&a;

p=p+3;

int i;

for (i=0;i<4;i++)

{

low=(\*p)&0x0f;

if(low<10)

low=low+'0';

else

low=low+'A'-10;

high=(\*p >> 4)&0x0f;

if (high<10)

high=high+'0';

else

high=high+'A'-10;

printf("%c %c\n",high,low);

p--;

}

}

（2）

#include <stdio.h>

#define N 80

void DeleteSpace (char \*p[],int n);

int main(void)

{

int i,j,n;

printf( "input the number(s) of the line(s):\n" );

scanf("%d",&n);

n=n+1;

char a[n][N],\*p[n];

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

gets(a[i]);

p[i]=a[i];

}

change(p,n);

return 0;

}

void change(char \*p[],int n)

{

int i,j,k,t;

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

for(j=0;p[i][j]!='\0';j++){

if(p[i][j]==' '&&p[i][j+1]==' '){

k=j+1;

while(p[i][k++]==' ');

k--;

t=j+1;

while(p[i][t++]=p[i][k++]);

}

}

if(strlen(p[i])>0)

printf("%s\n",p[i]);

}

}

（3）#include<stdio.h>

#include<string.h>

void down\_sort( int \* , int ) ;

void up\_sort( int \* , int ) ;

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

{

int n , i ;

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

printf( "请输入要输入整数的个数：" ) ;

scanf( "%d" , &n ) ;

int a[n] ;

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

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

}

if( argc > 1 && !strcmp(argv[1] , "-d") )

p = down\_sort ;

else

p = up\_sort ;

p( a , n ) ;

printf( "排序后：\n" ) ;

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

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

}

return 0 ;

}

void down\_sort( int \*a , int m )

{

int i , j , t ;

for( i = 0 ; i < m - 1 ; i++ ){

for( j = 0 ; j < m-1-i ; j++ ){

if( \*(a+j) < \*(a+j+1) ){

t = \*(a+j) ;

\*(a+j) = \*(a+j+1) ;

\*(a+j+1) = t ;

}

}

}

}

void up\_sort( int \*a , int m )

{

int i , j , t ;

for( i = 0 ; i < m - 1 ; i++ ){

for( j = 0 ; j < m-1-i ; j++ ){

if( \*(a+j) > \*(a+j+1) ){

t = \*(a+j) ;

\*(a+j) = \*(a+j+1) ;

\*(a+j+1) = t ;

}

}

}

}

（4）#include<stdio.h>

#define M 2

#define N 2

int main(void)

{

char courses[M][30] , students[N][20] ;

float tables[N][M] ;

int c , s ;

char \* co = &courses[0][0] ;

char \* st = &students[0][0] ;

float \* gr = &tables[0][0] ;

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

{

printf( "请输入第 %d 门课程的名字：\n" , c+1 ) ;

scanf( "%s" , courses[c] ) ;

}

for( s = 0 ; s < N ; s++ )

{

printf( "请输入第 %d 个学生的名字\n" , s+1 ) ;

scanf( "%s" , students[s] ) ;

}

for( s = 0 ; s < N ; s++ )

{

printf( "对于学生 %s ：\n" , students[s] ) ;

printf( "输入他的成绩:\n" ) ;

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

{

printf( "请输入 %s 的成绩： \n" , courses[c] ) ;

scanf( "%f" , &tables[s][c] ) ;

}

}

float sum = 0 ;

for( s = 0 ; s < N ; s++ )

{

sum = 0 ;

printf( " %s 的各门课程的平均成绩为:\n" , st+s\*20 ) ;

for( c = 0 ; c < M ; c++ ) sum += \* (gr+s\*M+c);

printf( "%f\n" , sum/M ) ;

}

float sum\_ = 0 ;

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

{

sum\_ = 0 ;

printf( "对于 %s ，全班的平均成绩为:\n" , co+c\*30 ) ;

for( s = 0 ; s < N ; s++ )sum\_ += \* (gr+s\*M+c) ;

printf( "%f\n" , sum\_/N ) ;

}

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

{

sum\_ = 0;

printf( "低于 %s 课程平均分的人数为：\n" , co+c\*30 ) ;

int cou = 0 ;

for( s = 0 ; s < N ; s++ )

sum\_ += \* (gr+s\*M+c);

for( s = 0 ; s < N ; s++ )

{

if( \*(gr+s\*M+c) < sum\_/N )cou++ ;}

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

cou = 0 ; }

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

{

printf( "对于 %s 课程，全班不及格人数为：\n" , co+c\*30 ) ;

int cou = 0 ;

for( s = 0 ; s < N ; s++ )

{

if( \*(gr+s\*M+c) < 60 )cou++ ; }

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

cou = 0 ; }

for( c = 0 ; c < M ; c++ ){

printf( "对于课程 %s ,全班90分(含90分)以上的人数为:\n",co+c\*30;

int cou = 0 ;

for( s = 0 ; s < N ; s++ )

{

if( \*(gr+s\*M+c) >= 90 ) cou++ ; }

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

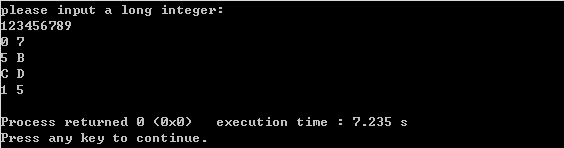
cou = 0 ; }

return 0 ;

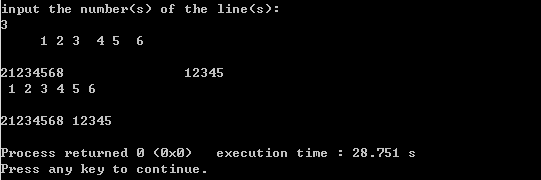
}

【测试】

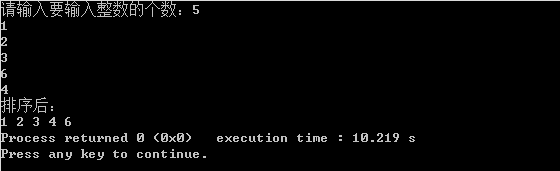
（1）结果正确：



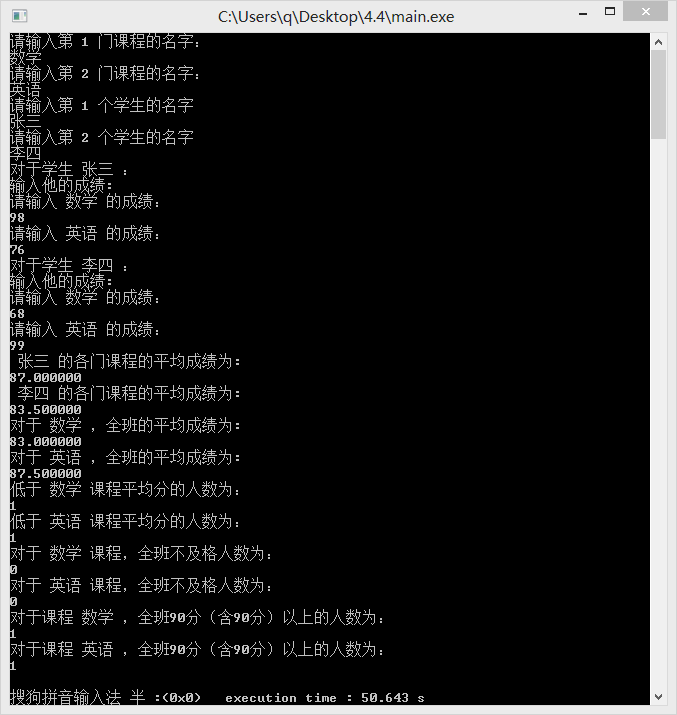
（2）结果正确：



（3）结果正确：



（4）结果正确：



6.3 实验体会

源程序改错中，依赖于平时良好的输入习惯，在输入过程中已经可以检查出大多数语法错误，或者拼写错误。接下来再检查一下逻辑是否有误，确认最终无误即可。

源程序修改与替换题锻炼我们将充分利用指针的能力，同时要注意指针的含义，指针指向对象的含义，充分理解防止出现问题，也为以后多重引用奠定基础。

程序设计题则是自主的设计实践，根据题目要求自主设计指针，增强对于所学知识的熟练度和理解程度，锻炼编写代码的能力以及并通过问题的进一步引申锻炼思维的灵活性。

**7结构与联合实验**

7.1实验目的

（1）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）掌握动态存储分配函数的用法，掌握自引用结构和单向链表的创建、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实验内容及要求

7.2.1. 表达式求值的程序验证

设有说明：

char u[]="UVWXYZ";

char v[]="xyz";

struct T{

int x;

char c;

char \*t;

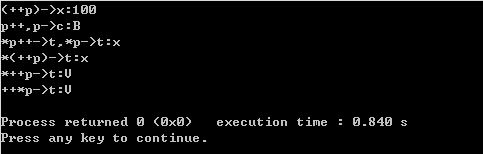
}a[]={{11,'A',u},{100,'B',v}},\*p=a;

请先自己计算表2.1中表达式的值，然后编写程序并运行来加以验证。（各表达式相互无关）

**表2.1 表达式值的计算**

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

表达式的计算值见图表。程序运行结果正确。



7.2.2 源程序修改替换

下面所给源程序的功能是：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。请完成以下工作：

（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。

（2）修改替换creat\_list函数，将其建成一个后进先出的链表。后进先出的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。

**程序代码：**

#include ”stdio.h”

#include ”stdlib.h”

struct s\_list{

int data;

struct s\_list \*next;

};

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

int main(void) {

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

int s[] = {1, 2, 3, 4, 5, 6, 7, 8, 0};

creat\_list(head, s);

p = head;

while(p) {

printf("%d\t", p -> data);

p = p -> next; }

printf("\n");

return 0;

}

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

{

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

if(p[0] == 0)

;

else {

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

loc\_head -> data = \*p++;

tail = loc\_head;

while(\*p) {

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

tail = tail -> next;

tail -> data = \*p++;}

tail -> next = NULL; }

headp = loc\_head;

}

【分析】（1）在函数creat\_list中，需要被操作的是头指针的地址值，所以参数类型不应该是struct s\_list \*headp而是struct s\_list \*\*headp，同样的在引用和赋值的时候也需要相应的改变类型。

（2）先进后出的链表不变，和（1）相同。改变后续链表指向实现题目要求。

【程序】

（1）

#include<stdio.h>

#include<stdlib.h>

struct s\_list{

int data;

struct s\_list \*next;

};

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

int main(void) {

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

int s[] = {1, 2, 3, 4, 5, 6, 7, 8, 0};

creat\_list(&head, s);

p = head;

while(p) {

printf("%d\t", p -> data);

p = p -> next; }

printf("\n");

return 0;

}

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

{

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

if(p[0] == 0)

;

else{

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

loc\_head -> data = \*p++;

tail = loc\_head;

while(\*p) {

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

tail = tail -> next;

tail -> data = \*p++; }

tail -> next = NULL; }

\*headp = loc\_head;

}

（2）

#include <stdio.h>

#include <stdlib.h>

struct s\_list{

int data;

struct s\_list \*next;

};

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

int main()

{

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

int s[]={1,2,3,4,5,6,7,8,0};

creat\_list(&head,s);

p=head;

while(p){

printf("%d\t",p->data);

p=p->next; }

printf("\n");

return 0;

}

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

{

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

struct s\_list\*temp;

if(p[0]==0 ;

else {

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

loc\_head->data=\*p++;

tail=loc\_head;

while(\*p){

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

temp->next=loc\_head;

loc\_head=temp;

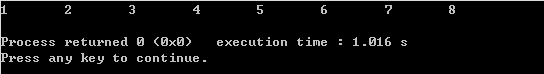
loc\_head->data = \*p++; }

tail->next=NULL; }

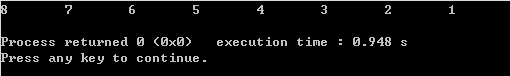
\*headp=loc\_head;

}

【测试】（1）程序运行结果如下，程序正确。



（2）程序运行结果如下，程序正确。



7.2.3 程序设计

编写并上机调试运行能实现以下功能的程序：

（1）编写一个程序，实现以下功能：定义一个字段结构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bit***i***（***i*** = 0, 1,……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);

}

（2）假设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，试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（保留两位小数）。

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。

【分析】

（1）依照题意设置字段结构和函数，优先级可以通过函数出现位置来实现。其他不变，照常输入。

（2）设置结构，分别输入所需要的六项数据，储存，设计菜单实现功能选择。

【程序】

（1）

#include<stdio.h>

struct bits{

unsigned char bit0:1, bit1:1, bit2:1, bit3:1, bit4:1, bit5:1, bit6:1, bit7:1;

}a;

union w{

unsigned char t;

struct bits a;

}m;

void f0(struct bits b) {

printf("the 1 function %d is called!\n", b.bit0); }

void f1(struct bits b) {

printf("the 2 function %d is called!\n", b.bit1);}

void f2(struct bits b) {

printf("the 3 function %d is called!\n", b.bit2); }

void f3(struct bits b){

printf("the 4 function %d is called!\n", b.bit3);}

void f4(struct bits b){

printf("the 5 function %d is called!\n", b.bit4);}

void f5(struct bits b){

printf("the 6 function %d is called!\n", b.bit5);}

void f6(struct bits b){

printf("the 7 function %d is called!\n", b.bit6);}

void f7(struct bits b){

printf("the 8 function %d is called!\n", b.bit7);}

int main()

{

void (\*p\_fun[8])(struct bits b);

printf("input char: ");

m.t=getchar();

m.a.bit0 = 1;

p\_fun[0] = f0;

p\_fun[1] = f1;

p\_fun[2] = f2;

p\_fun[3] = f3;

p\_fun[4] = f4;

p\_fun[5] = f5;

p\_fun[6] = f6;

p\_fun[7] = f7;

if(m.a.bit0) p\_fun[0](m.a);

if(m.a.bit1) p\_fun[1](m.a);

if(m.a.bit2) p\_fun[2](m.a);

if(m.a.bit3) p\_fun[3](m.a);

if(m.a.bit4) p\_fun[4](m.a);

if(m.a.bit5) p\_fun[5](m.a);

if(m.a.bit6) p\_fun[6](m.a);

if(m.a.bit7) p\_fun[7](m.a);

return 0;

}

（2）

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct stu {

long num;

char s[10];

int c1;

int c2;

int c3;

int c4;

struct stu \*next;

};

void creatlist(struct stu \*\*headp);

void outputlist(struct stu \*\*headp);

void correctlist(struct stu \*\*headp);

void avstu(struct stu \*\*headp);

void sumstu(struct stu \*\*headp);

int main()

{

struct stu \*head=NULL;

creatlist(&head);

outputlist(&head);

correctlist(&head);

avstu(&head);

sumstu(&head);

return 0;

}

void creatlist(struct stu \*\*headp)

{

\*headp=(struct stu \*)malloc(sizeof(struct stu));

struct stu \*tail=\*headp,\*p=\*headp;

printf("please input the following item devided by square:\n");

printf("ID\tname\tEnglish grade\tmath grade\tphysics grade\tC language grade");

printf("end with 0\n");

for(;;){

scanf("%ld",&p->num);

if(p->num==0)break;

scanf("%s %d %d %d %d",p->s,&p->c1,&p->c2,&p->c3,&p->c4);

p=(struct stu \*)malloc(sizeof(struct stu));

tail->next=p;

tail=p; }

tail->next=NULL;

}

void outputlist(struct stu \*\*headp)

{

struct stu \*p=\*headp;

while(p->next!=NULL) {

printf("ID %ld NAME %s ENG %d MATH %d PHY %d C %d\n",p->num,p->s,p->c1,p->c2,p->c3,p->c4);

p=p->next; }

}

void correctlist(struct stu \*\*headp)

{

char c;

struct stu \*p;

p=\*headp;

printf("1:look up by name\n");

printf("2:look up by number\n");

c=getchar();

c=getchar();

switch(c){

case '1':

printf("please input the name\n");

char t[10];

scanf("%s",t);

for(;p->next!=NULL&&strcmp(p->s,t);p=p->next)

;

if(!strcmp(p->s,t))

printf("FIND ID %ld NAME %s ENG %d MATH %d PHY %d C %d\n",p->num,p->s,p->c1,p->c2,p->c3,p->c4);

else if(p->next==NULL) {

printf("NOT FOUND\n");

return; }

break;

case '2':

printf("please input the number\n");

long tmp;

scanf("%ld",&tmp);

for(;p->next!=NULL&&tmp!=p->num;p=p->next) ;

if(tmp==p->num)

printf("FIND ID %ld NAME %s ENG %d MATH %d PHY %d C %d\n",p->num,p->s,p->c1,p->c2,p->c3,p->c4);

else if(p->next==NULL) {

printf("NOT FOUND\n");

return;}

break; }

printf("which one to change?\n");

printf("1:number\n");

printf("2:name\n");

printf("3:Grade of English\n");

printf("4:Grade of Math\n");

printf("5:Grade of Physics\n");

printf("6:Grade of C Language\n");

printf("7:I don't want to chage anything\n");

c=getchar();

c=getchar();

printf("please reinput\n");

switch(c) {

case '1':scanf("%ld",&p->num);break;

case '2':scanf("%s",p->s);break;

case '3':scanf("%d",&p->c1);break;

case '4':scanf("%d",&p->c2);break;

case '5':scanf("%d",&p->c3);break;

case '6':scanf("%d",&p->c4);

default :return; }

}

void avstu(struct stu \*\*headp)

{

struct stu \*p=\*headp;

for(;p->next!=NULL;p=p->next) {

double tp;

tp=(p->c1+p->c2+p->c3+p->c4)/4.0;

printf("ID %ld Name %s average %.2lf\n",p->num,p->s,tp); }

}

void sumstu(struct stu \*\*headp)

{

struct stu \*p;

p=\*headp;

for(;p->next!=NULL;p=p->next) {

double tp;

int sum;

sum=p->c1+p->c2+p->c3+p->c4;

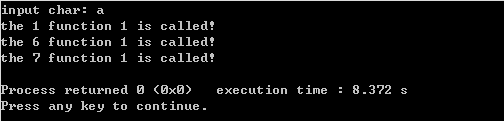
tp=sum/4.0;

printf("ID %ld Name %s sum is %d average is %.2lf\n",p->num,p->s,sum,tp); }

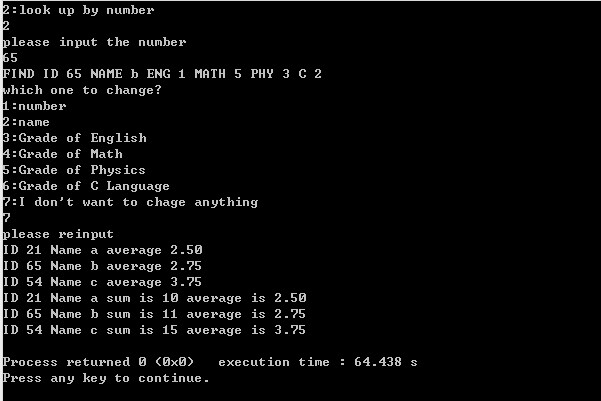
}

【测试】

（1）结果正确：



（2）结果正确：



7.3 实验体会

程序验证题旨在检验我们对指针和结构的了解程度，先自己计算然后再用程序进行验算，在验算过程中需要注意初始化指针位置，防止前面的运算对后续运算结果产生影响，而且每一次运算之前都要初始化。

源程序修改与替换题进一步锻炼我们对指针的理解能力，同时要注意引用的时候引用类型时什么，严谨杜绝副作用产生逻辑错误。

程序设计题则是自主的设计实践，根据题目要求自主设计链表，增强对于所学知识的熟练度和理解程度，锻炼编写代码的能力以及并通过问题的进一步引申锻炼思维的灵活性。

**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）熟悉掌握流式文件的读写方法。

8.2实验内容及要求

8.2.1. 文件类型的程序验证

设有程序：

#include <stdio.h>

int main()

{

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,fp2);

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;

}

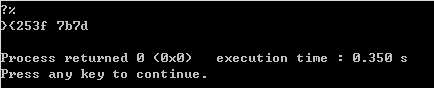
（1）请思考程序的输出结果，然后通过上机运行来加以验证。

（2）将两处sizeof(short)均改为sizeof(char)后，结果有什么不同？为什么？

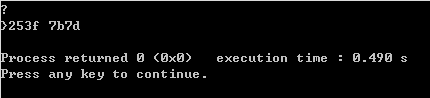
（3）将fprintf(fp2,”%hx%hx”,a,b)改为fprintf(fp2,”%d%d”,a,b)后结果有什么不同。

【解答】

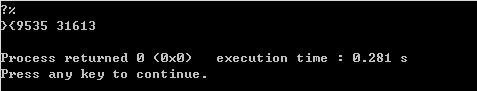
（1）上机运行结果如图，结果正确。



（2）结果如图。原因：由于类型的变化，相当于一个字符型指针去读取一个数字，会按照每个字节分割，而不是作为一个短整型数据两个字节一起输出。



（3）结果如图。按照十进制输出。



8.2.2 源程序修改替换

已知将制定文本文件内容在屏幕上显示出来的命令行格式为

type filename

（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，使之能够正确的完成指定任务。

**程序代码：**

#include <stdio.h>

#include <stdlib.h>

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

{

char ch;

FILE \*fp;

if(argc!=2){

printf("Argument error!\n");

exit(-1);

}

if((fp=fopen(argv[1],"r"))==NULL){ /\*fp指向filename\*/

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

exit(-1);

}

while(ch=fgetc(fp)!=EOF) /\*从filename中读字符\*/

putchar(ch); /\*向显示器中写字符\*/

fclose(fp); /\*关闭filename\*/

return 0;

}

（2）用输入输出重定向freopen改写上述源程序中的main函数。

【程序】

（1）

#include <stdio.h>

#include <stdlib.h>

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

{

char ch;

FILE \*fp;

if(argc!=2){

printf("Argument error!\n");

exit(-1);

}

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

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

exit(-1);

}

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

putchar(ch);

fclose(fp);

return 0;

}

（2）

#include <stdio.h>

#include <stdlib.h>

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

{

char ch;

FILE \*fp;

if(argc!=2){

printf("Argument error!\n");

exit(-1);

}

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

printf("Can'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

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。

【程序】

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

int main()

{

FILE \*fp;

char a[40];

int i=0;

fp=fopen("test.txt","w");

gets(a);

a[0]=toupper(a[0]);

while(\*(a+i)!='\0'){

if(islower(\*(a+i))&&\*(a+i-1)==' ')

\*(a+i)=toupper(\*(a+i));

i++;

}

puts(a);

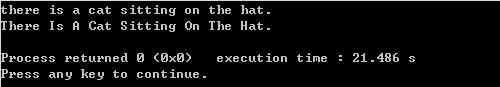
fputs(a,fp);

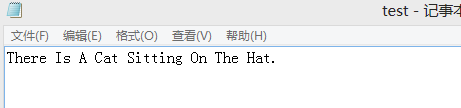
fclose(fp);

return 0;

}

【测试】如下图，结果正确：





8.3 实验体会

这一部分感觉有些困难，尤其是如果要统计大量数据的话。写了好多代码还是没有太大手感，继续努力吧。