**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。

**源程序：**

#include<stdio.h>

int main(void)

{

float \*p;

scanf("%f",p);

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

return 0;

}

源程序分析及改正：本程序中一共存在1处错误，分析结果如下：

1. scanf("%f",p);

**【分析】**对于一个未初始化的指针进行操作，有可能导致内存溢出。

**【改正】**在本句前加p = (float \*)malloc(sizeof(float));并加上stdlib库。

**6.2.2．**源程序完善、修改、替换

(1) 下面源程序的功能是,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。

**源程序：**

#include<stdio.h>

#include <string.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:

goto down;

}

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);

down:

return 0;

}

(2) 请上机运行第（1）题程序，使之能按要求输出下面结果（注：（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.

**【分析】**第一个横线上要声明后面要使用的函数指针p，所以为char (\*p)(char \*,const char \*);。getchar();后面的两个横线要求的是输入两个字符串，第一个是a，第二个是b，所以分别为gets(a);和gets(b);最后一条横线是选择执行不同的功能，所以为p。

**【程序】**

#include<stdio.h>

#include <string.h>

int main(void)

{

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

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);

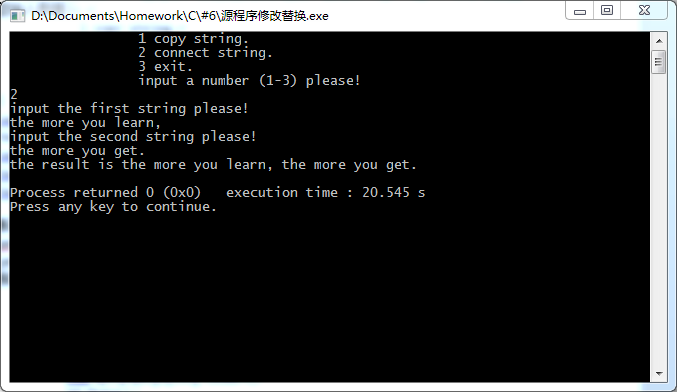
printf("the result is %s\n",result);

down:

return 0;

}

**【测试】**输入与第二题要求一致



**6.2.3．跟踪调试**源程序

请按下面的要求对所给源程序进行操作，并回答问题和排除错误。

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

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

**源程序：**

#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++)

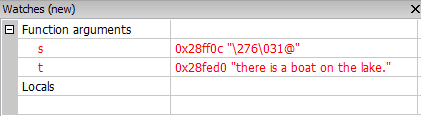
;

return (s);

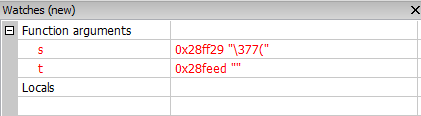
}

**【跟踪】**

进入strcpy时，s如下图



返回main时s如下图



**【分析】**while后s为字符串的最后一位，不是首位，要将s返回。

**【程序】**

#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)

{

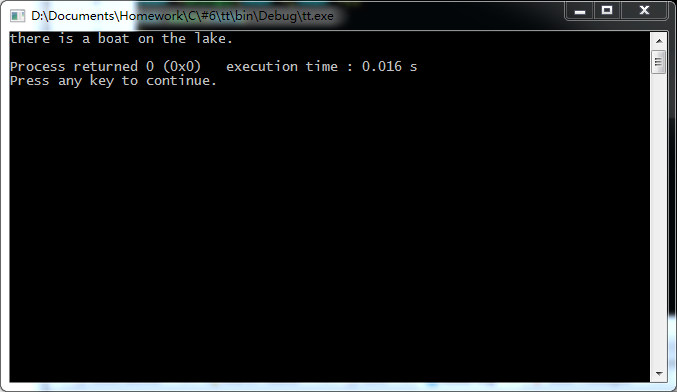
char \*temp=s;

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

return (temp);

}

**【测试】**修改后输出与预想一致



**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或函数：

(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位并以十六进制数字字符的形式进行显示。

(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-d决定，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。（main函数参数的处理见下面的第(三)大点）

(4) 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：

(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）

**【分析】**运用位处理“与”取出高四位和低四位

**【程序】**

#include<stdio.h>

int main(void){

long int a;

scanf("%ld",&a);

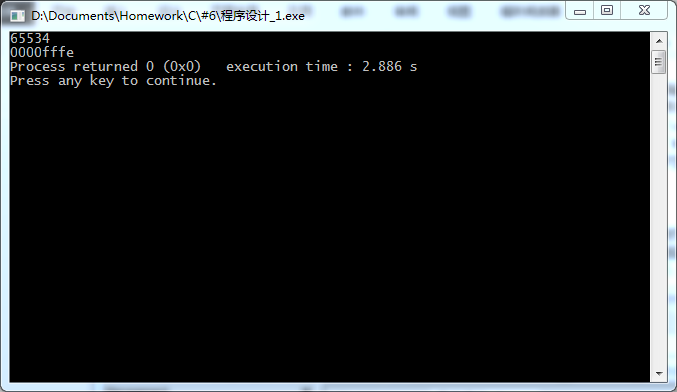
for(int i=0;i<8;i++)

printf("%x",(a&(0x0f<<(28-4\*i)))>>(28-4\*i));

return 0;

}

**【调试】**输出与预想一致



（2）

**【分析】**指针数组中的每一个指针指向一个字符串，再对指针数组进行处理即可

**【程序】**

#include<stdio.h>

#include<stdlib.h>

int main(void){

int n,i,j,c,c2;

scanf("%d",&n);

getchar();

char \*str[n];

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

str[i]=(char \*)malloc(sizeof(char)\*81);

gets(str[i]);

for(j=0,c=0;str[i][j+c]!='\0';j++){

str[i][j]=str[i][j+c];

if(str[i][j+c]==' '){

for(;str[i][j+c]==' ';c++);

c--;

}

}

str[i][j]='\0';

}

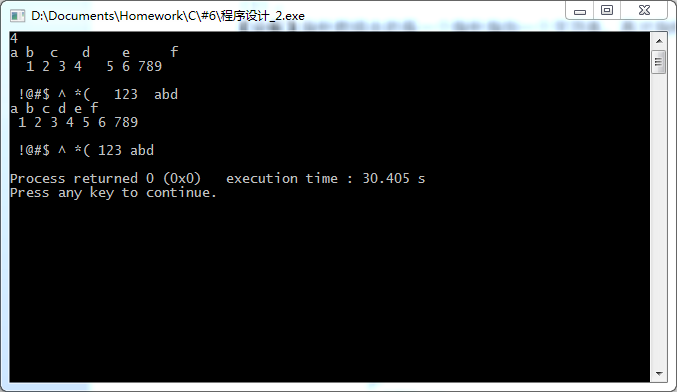
for(i=0;i<n;i++)

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

return 0;

}

**【调试】**输出与题目要求一致



（3）

**【分析】**分别写两个函数，一个递增，一个递减，通过判断主函数是否要参数来决定调用哪一个函数进行排序。

**【程序】**

#include<stdio.h>

#define MAX\_BUFF\_SIZE 65535

void m\_sort\_i(int \*buff,int len);

void m\_sort\_d(int \*buff,int len);

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

int buff[MAX\_BUFF\_SIZE];

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

f=(argc==1)?m\_sort\_i:m\_sort\_d;

int N;

scanf("%d",&N);

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

scanf("%d",buff+i);

f(buff,N);

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

printf("%5d",buff[i]);

return 0;

}

void m\_sort\_i(int \*buff,int len){

if(len==1)

return;

m\_sort\_i(buff,len/2);

m\_sort\_i(buff+len/2,len-len/2);

int temp[MAX\_BUFF\_SIZE],i=0,j=len/2,c=0;

for(;i<len/2&&j<len;c++)

temp[c]=(buff[i]<buff[j])?buff[i++]:buff[j++];

for(;i<len/2;i++,c++)

temp[c]=buff[i];

for(;j<len;j++,c++)

temp[c]=buff[j];

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

buff[c]=temp[c];

}

void m\_sort\_d(int \*buff,int len){

if(len==1)

return;

m\_sort\_d(buff,len/2);

m\_sort\_d(buff+len/2,len-len/2);

int temp[MAX\_BUFF\_SIZE],i=0,j=len/2,c=0;

for(;i<len/2&&j<len;c++)

temp[c]=(buff[i]>buff[j])?buff[i++]:buff[j++];

for(;i<len/2;i++,c++)

temp[c]=buff[i];

for(;j<len;j++,c++)

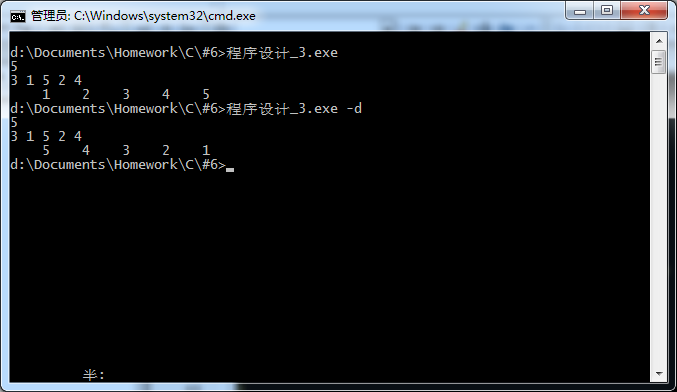
temp[c]=buff[j];

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

buff[c]=temp[c];

}

**【调试】**结果如图，符合题目要求



（4）

**【分析】**使用一个二维数组代表每个学生和其各门课程，再用指针的移动进行各个统计运算即可。用两个个数组存放那学生的平均成绩和各门课程的品均成绩会更方便。

**【程序】**

#include<stdio.h>

#include<stdlib.h>

#define N 5

#define M 5

int main(void){

int \*\*stu = (int \*\*)malloc(sizeof(int \*)\*N);

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

\*(stu+i) = (int \*)malloc(sizeof(int)\*M);

float stuavr[N],clsavr[M];

int \*\*one=stu;

float \*\*oneavr;

int i,j,k;

float sum;

for(;one<stu+N;one++)

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

scanf("%d",\*one+i);

for(one=stu;one<stu+N;one++){

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

sum+=\*(\*one+i);

printf("Stu #%d : Avr %f\n",(one-stu)/sizeof(int\*\*),\*(stuavr+(one-stu)/sizeof(int\*\*))=sum/M);

}

putchar('\n');

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

for(one=stu,sum=0;one<stu+N;one++)

sum+=\*(\*one+i);

printf("Cls #%d : Avr %f\n",i,\*(clsavr+i)=sum/N);

}

putchar('\n');

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

for(sum=0,j=0,k=0,one=stu;one<stu+N;one++){

sum += (\*(\*one+i)<\*(clsavr+i))?1:0;

j+=(\*(\*one+i)<60)?1:0;

k+=(\*(\*one+i)>=90)?1:0;

}

printf("Cls #%d : Blw Avr %d\n",i,(int)sum);

printf(" Blw 60 %d\n",i,j);

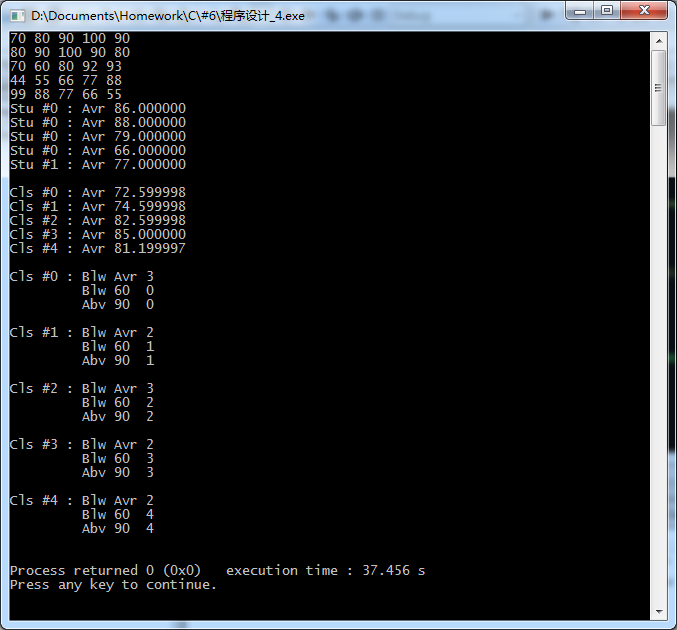
printf(" Abv 90 %d\n",i,k);

putchar('\n');

}

}

**【调试】**使用5个学生，每个学生5门课测试



结果与预想一致

**6.2.5．**选做题

编写并上机调试运行能实现以下功能的程序：

(1) 设有N位整数和M位小数（N=20，M=10）的数据a,b。编程计算a+b并输出结果。

如：12345678912345678912.1234567891 + 98765432109876543210.0123456789

(2) 编一个写使用复杂声明“char \*(\*p[2])(const char \*,const char \*);”的程序。

提示：p中元素可为strstr等函数名。

（1）

**【分析】**利用字符串逐位处理的高精度算法对于两个等长且小数点位置相同的字符串只需同时处理即可。

**【程序】**

#include<stdio.h>

int main(void){

char a[32]="12345678912345678912.1234567891";

char b[32]="98765432109876543210.0123456789";

int add=0,c=1,i;

char result[33],\*r;

memset(result,0,33);

for(i=30;i>=0;i--,c=1){

if(a[i]=='.')

result[i+c]='.',i--;

result[i+c]=(char)((a[i]+b[i]+add-2\*'0')%10+'0');

add=(a[i]+b[i]+add-2\*'0')/10;

}

result[32]='\0';

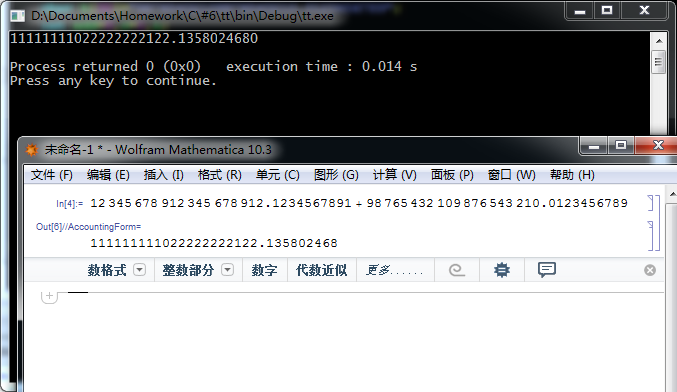
for(r=result;\*r=='\0';r++);

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

return 0;

}

**【调试】**输出结果与Mathematica计算结果一致



（2）

**【分析】**一个指针数组，它是函数指针数组，这个函数指针指向的函数函数的形参是两个const char \*类型值，返回值是char \*类型

**【程序】**

#include<stdio.h>

#include<string.h>

int main(void){

int c;

char str[2][65535];

char \*(\*p[2])(const char \*,const char \*);

p[0]=strcat,p[1]=strstr;

printf("Please input function\n1 for strcat\n2 for strstr\n0 to exit\n\n");

for(;;){

scanf("%d",&c);

getchar();

switch(c){

case 1:

printf("please input str1: \n");

gets(str[1]);

printf("please input str2: \n");

gets(str[2]);

printf("After string catenation, the result is: \n%s\n\n",p[0](str[1],str[2]));

break;

case 2:

printf("please input str1: \n");

gets(str[1]);

printf("please input str2: \n");

gets(str[2]);

printf("After substring search, the result is: \n%s\n\n",p[1](str[1],str[2])==NULL?"not found!":"found!");

break;

case 0:

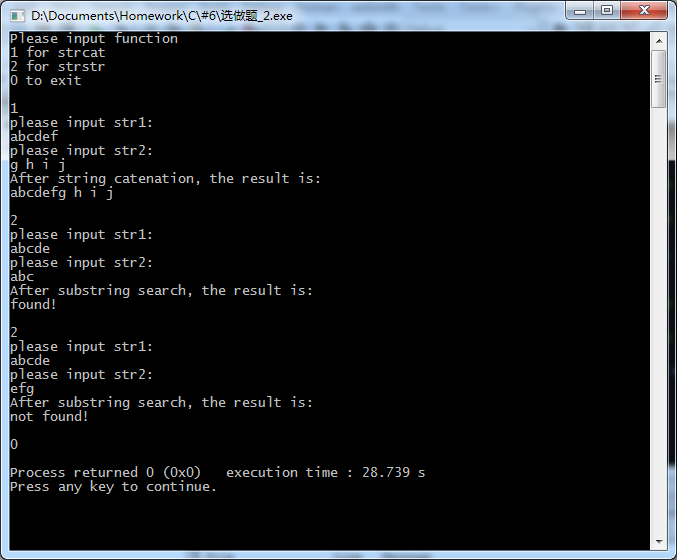
return 0;

}

}

}

**【调试】**测试所有功能，均正常



**(三) 指定main函数的参数**

选择“project/ set programs' arguments…”菜单命令，即可打开图2.12所示的对话框，在“Program arguments”文本框中输入main函数的参数。注意只输入命令行中文件名后的参数，文件名不输人。

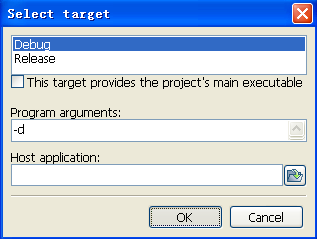
****

图2.12 输入main函数的参数

**6.3实验小结**

通过本次实验，我了解了指针的说明、赋值、使用。掌握了用指针引用数组的元素，熟悉指向数组的指针的使用。掌握了指针函数与函数指针的用法。特别是了解了带有参数的main函数的用法，这对程序的调试是极为有利的，如可以将一个程序的输出通过main函数传给另一个程序，增加了程序的一些功能。