

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 网安2003班**

**学 号： U202012068**

**姓 名： 叶礼亮**

**指导教师： 余林琛**

**报告日期： 2020/10/17**

**网络空间安全学院**

**目 录**

[**1 表达式和标准输入输出实验 1**](#_Toc404837920)

[1.1 实验目的 1](#_Toc404837921)

[1.2 实验内容 1](#_Toc404837922)

[1.3 实验小结 1](#_Toc404837923)

[**2 流程控制实验 2**](#_Toc404837924)

[2.1 实验目的 1](#_Toc404837921)

[2.2 实验内容 1](#_Toc404837922)

[2.3 实验小结 1](#_Toc404837923)

[**3 函数与程序结构实验 3**](#_Toc404837929)

[3.1 实验目的 1](#_Toc404837921)

[3.2 实验内容 1](#_Toc404837922)

[3.3 实验小结 1](#_Toc404837923)

[**4 编译预处理实验 4**](#_Toc404837934)

[4.1 实验目的 1](#_Toc404837921)

[4.2 实验内容 1](#_Toc404837922)

[4.3 实验小结 1](#_Toc404837923)

[**5 数组实验 5**](#_Toc404837938)

[5.1 实验目的 1](#_Toc404837921)

[5.2 实验内容 1](#_Toc404837922)

[5.3 实验小结 1](#_Toc404837923)

[**6 指针实验 6**](#_Toc404837943)

[6.1 实验目的 1](#_Toc404837921)

[6.2 实验内容 1](#_Toc404837922)

[6.3 实验小结 1](#_Toc404837923)

[**7 结构与联合实验 7**](#_Toc404837948)

[7.1 实验目的 1](#_Toc404837921)

[7.2 实验内容 1](#_Toc404837922)

[7.3 实验小结 1](#_Toc404837923)

[**8 文件实验 8**](#_Toc404837953)

[8.1 实验目的 1](#_Toc404837921)

[8.2 实验内容 1](#_Toc404837922)

[8.3 实验小结 1](#_Toc404837923)

[**参考文献 9**](#_Toc404837957)

# 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**

在下面所给的源程序中，函数strcopy(t, s)的功能是将字符串s复制给字符串t，并且返回串t的首地址。请单步跟踪程序，根据程序运行时出现的现象或观察到的字符串的值，分析并排除源程序的逻辑错误，使之能按照要求输出如下结果：

Input a string:

programming↙ （键盘输入）

programming

Input a string again:

language↙ （键盘输入）

language

1 #include<stdio.h>

2 char \*strcopy(char \*, const char \*);

3 int main(void)

4 {

5 char \*s1, \*s2, \*s3;

6 printf("Input a string:\n", s2);

7 scanf("%s", s2);

8 strcopy(s1, s2);

9 printf("%s\n", s1);

10 printf("Input a string again:\n", s2);

11 scanf("%s", s2);

12 s3 = strcopy(s1, s2);

13 printf("%s\n", s3);

14 return 0;

15 }

16

17 /\*将字符串s复制给字符串t，并且返回串t的首地址\*/

18 char \* strcopy(char \*t, const char \*s)

19 {

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

21 return (t);

22 }

**解答：**

（1）错误修改：

1) 第5行指针应当指向具体的内存，正确形式为：

char str1[20],str2[20],str3[20];

s1=str1,s2=str2,s3=str3;

2) 第6行和12行printf不用s2，正确形式为：

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

printf("Input a string again:\n");

3) 第21行应该返回字符串原始的开始地址，正确形式为：

char \*tmp=t; //先暂存串t开头的位置来返回

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

return (tmp);

（2）错误修改后运行结果：

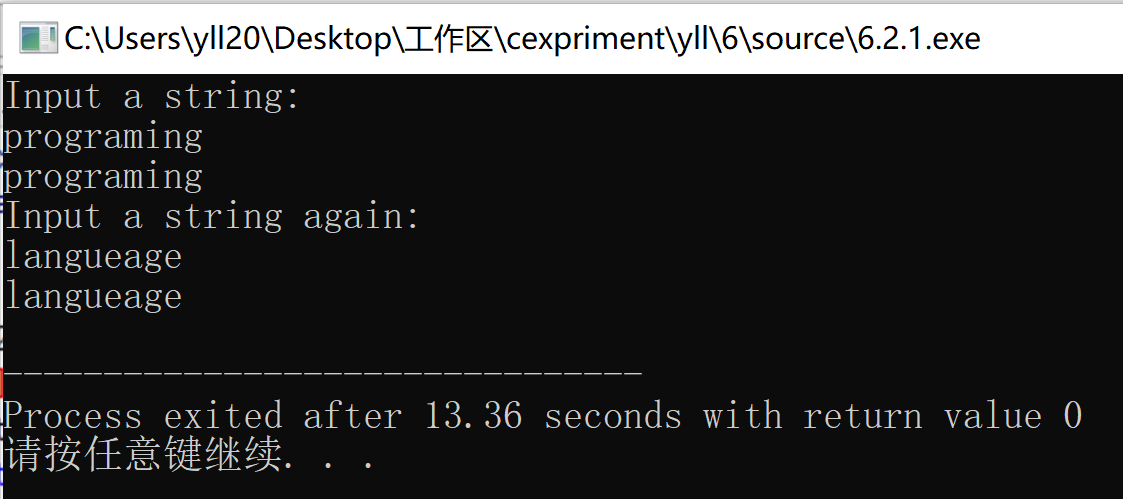
****

图6-1 改错题1的运行结果

**6.2.2 源程序完善、修改替换题**

（1）下面程序中函数strsort用于对字符串进行升序排序，在主函数中输入N个字符串存入通过malloc动态分配的存储空间，然后调用strsort对这N个串按字典序升序排序。

①请在源程序中的下划线处填写合适的代码来完善该程序。

#include<stdio.h>

#include<\_\_\_\_\_\_\_\_\_>

#include<string.h>

#define N 4

/\*对指针数组s指向的size个字符串进行升序排序\*/

void strsort(char \*s[], int size)

{

\_\_\_\_\_\_\_temp;

int i, j;

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

for (j=0; j<size-j-1; j++)

if (\_\_\_\_\_\_\_\_\_\_\_)

{

temp = s[j];

\_\_\_\_\_\_\_\_\_\_\_;

s[j+1] = temp;

}

}

int main()

{

int i;

char \*s[N], t[50];

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

{

gets(t);

s[i] = (char \*)malloc(strlen(t)+1);

strcpy(\_\_\_\_\_\_\_);

}

strsort(\_\_\_\_\_\_\_\_);

for (i=0; i<N; i++) puts(s[i]);

return 0;

}

②数组作为函数参数其本质类型是指针。例如，对于形参char \*s[]，编译器将其解释为char \*\*s，两种写法完全等价。请用二级指针形参重写strsort函数，并且在该函数体的任何位置都不允许使用下标引用。

**解答：**

1. 替换后的程序如下所示：

#include<stdio.h>

#include<stdlib.h> //malloc在stdlib里面

#include<string.h>

#define N 4

/\*对指针数组s指向的size个字符串进行升序排序\*/

void strsort(char \*s[], int size)

{

char \*temp; //从后面来看temp是指针

int i, j;

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

{

for (j=0; j<size-i-1; j++) //此处应为j<size-i-1;

if (strcmp(s[j],s[j+1])>0) //升序排序

{

temp = s[j];

s[j] = s[j+1];

s[j+1] = temp;

}

}

}

int main()

{

int i;

char \*s[N], t[50];

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

{

gets(t);

s[i] = (char \*)malloc(strlen(t)+1);

strcpy(s[i],t); //此处将t存入新开辟的空间

}

strsort(s,i); //此处调用排序函数

for (i=0; i<N; i++) puts(s[i]);

return 0;

}

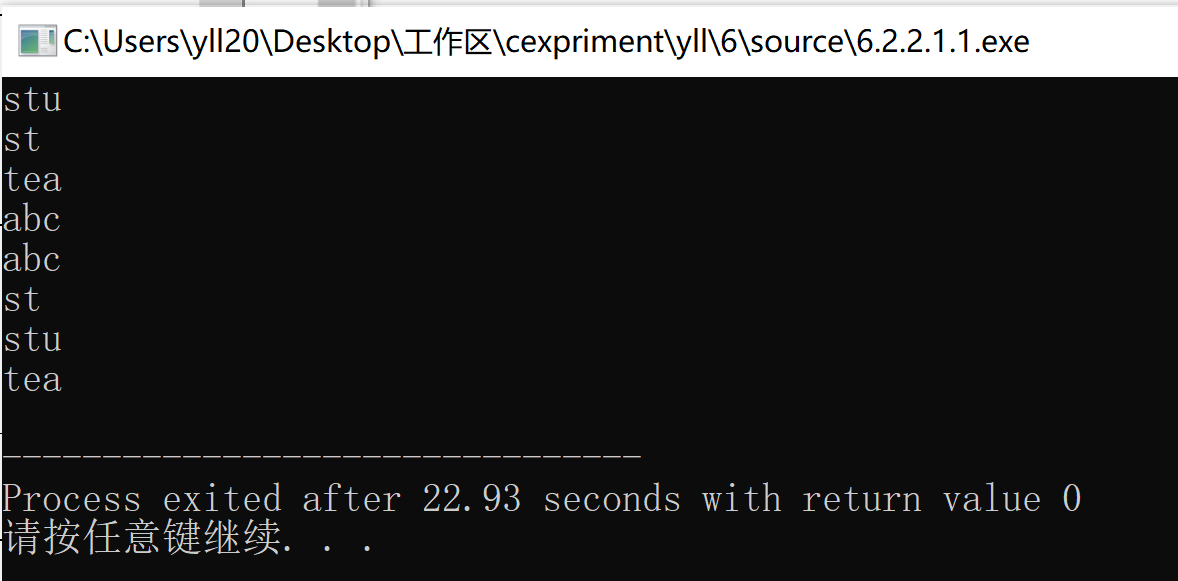


图6-2 程序替换题1的运行结果

1. 替换后的程序如下所示：

#include<stdio.h>

#include<stdlib.h> //malloc在stdlib里面

#include<string.h>

#define N 4

/\*对指针数组s指向的size个字符串进行升序排序\*/

void strsort(char \*\*s, int size)

{

char \*temp;

int i, j;

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

{

for (j=0; j<size-j-1; j++)

if (strcmp(\*(s+j),\*(s+j+1))>0)

{

temp = \*(s+j);

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

\*(s+j+1) = temp;

}

}

}

int main()

{

int i;

char \*s[N], t[50];

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

{

gets(t);

s[i] = (char \*)malloc(strlen(t)+1);

strcpy(s[i],t); //此处将t存入新开辟的空间

}

strsort(s,i+2); //此处调用排序函数

for (i=0; i<N; i++) puts(s[i]);

return 0;

}

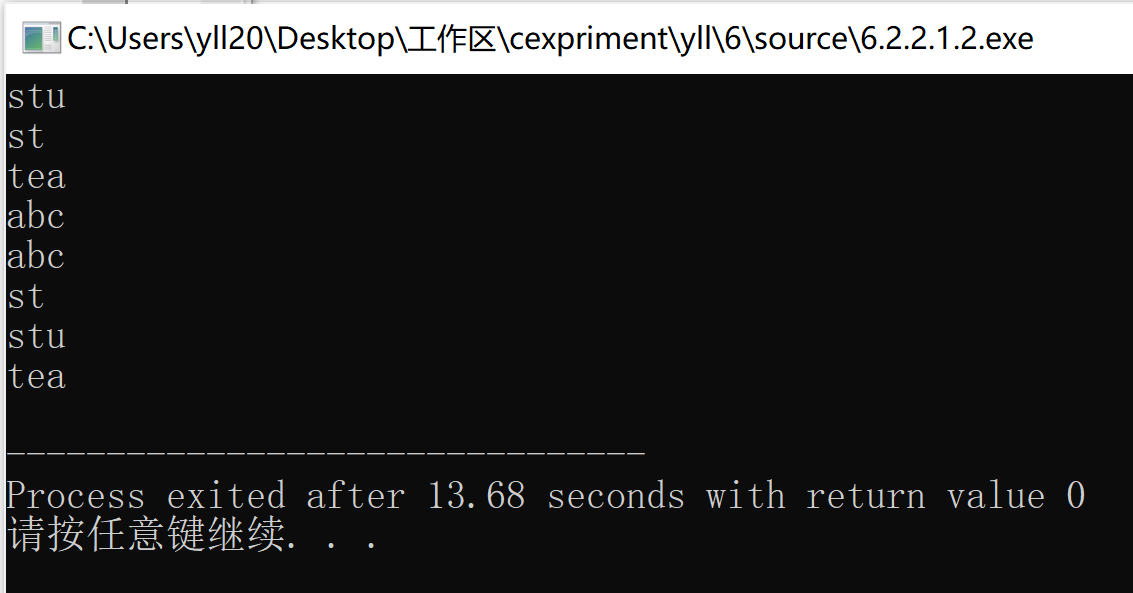


图6-3 程序替换题1的运行结果

（2）下面源程序通过函数指针和菜单选择来调用库函数实现字符串操作；串复制strcpy、串连接strcat或串分解strtok。

①请在源程序中的下划线处填写合适的代码来完善该程序，使之能按照要求输出下面结果：

1 copy string.

2 connect string.

3 parse string.

4 exit.

input a number (1-4) please!

2↙ （键盘输入）

input the first string please!

the more you learn,↙ （键盘输入）

input the second string please!

the more you get. ↙ （键盘输入）

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

# include<stdio.h>

# include<string.h>

int main (void)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

char a[80], b[80], \*result;

int choice;

while(1)

{

do

{

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

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

printf("\t\t3 parse string.\n");

printf("\t\t4 exit.\n");

printf("\t\tinput a number (1-4) please.\n");

scanf("%d", &choice);

}while(choice<1 || choice>4);

switch(choice)

{

case 1: p = strcpy; break;

case 2: p = strcat; break;

case 3: p = strok; break;

case 4: p = goto down;

}

getchar();

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

result = \_\_\_\_\_\_\_\_\_\_\_(a, b);

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

}

down:

return 0;

}

②函数指针的一个用途是用户散转程序，即通过一个转移表（函数指针数组）来实现多分枝函数处理，从而省去了大量的if语句或者switch语句。转移表中存放了各个函数的入口地址（函数名），根据条件的设定来查表选择执行相应的函数。请使用转移表而不是switch语句重写以上程序。

**解答：**

1. 替换后的程序如下所示：

# include<stdio.h>

# include<string.h>

int main (void)

{

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

char a[80], b[80], \*result;

int choice;

while(1)

{

do

{

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

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

printf("\t\t3 parse string.\n");

printf("\t\t4 exit.\n");

printf("\t\tinput a number (1-4) please.\n");

scanf("%d", &choice);

}while(choice<1 || choice>4);

switch(choice)

{

case 1: p = strcpy; break;

case 2: p = strcat; break;

case 3: p = strtok; break;

case 4: goto down;

}

getchar();

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

gets(a); //读a

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

gets(b); //读b

result = p(a, b); //干活

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

}

down:

return 0;

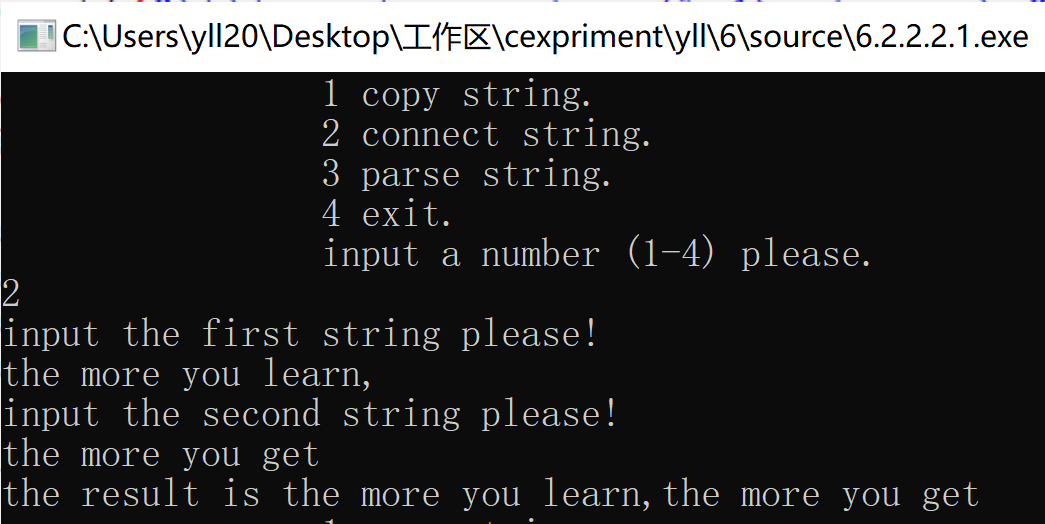
} 

图6-4 程序替换题2的运行结果

1. 替换后的程序如下所示：

# include<stdio.h>

# include<string.h>

int main (void)

{

char \*(\*p[4])(char \*a,char \*b);

p[1]=strcpy;

p[2]=strcat;

p[3]=strtok;

char a[80], b[80], \*result;

int choice;

while(1)

{

do

{

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

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

printf("\t\t3 parse string.\n");

printf("\t\t4 exit.\n");

printf("\t\tinput a number (1-4) please.\n");

scanf("%d", &choice);

}while(choice<1 || choice>4);

if(choice==4) goto down;

getchar();

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

gets(a); //读a

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

gets(b); //读b

result = p[choice](a, b); //干活

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

}

down:

return 0;

}

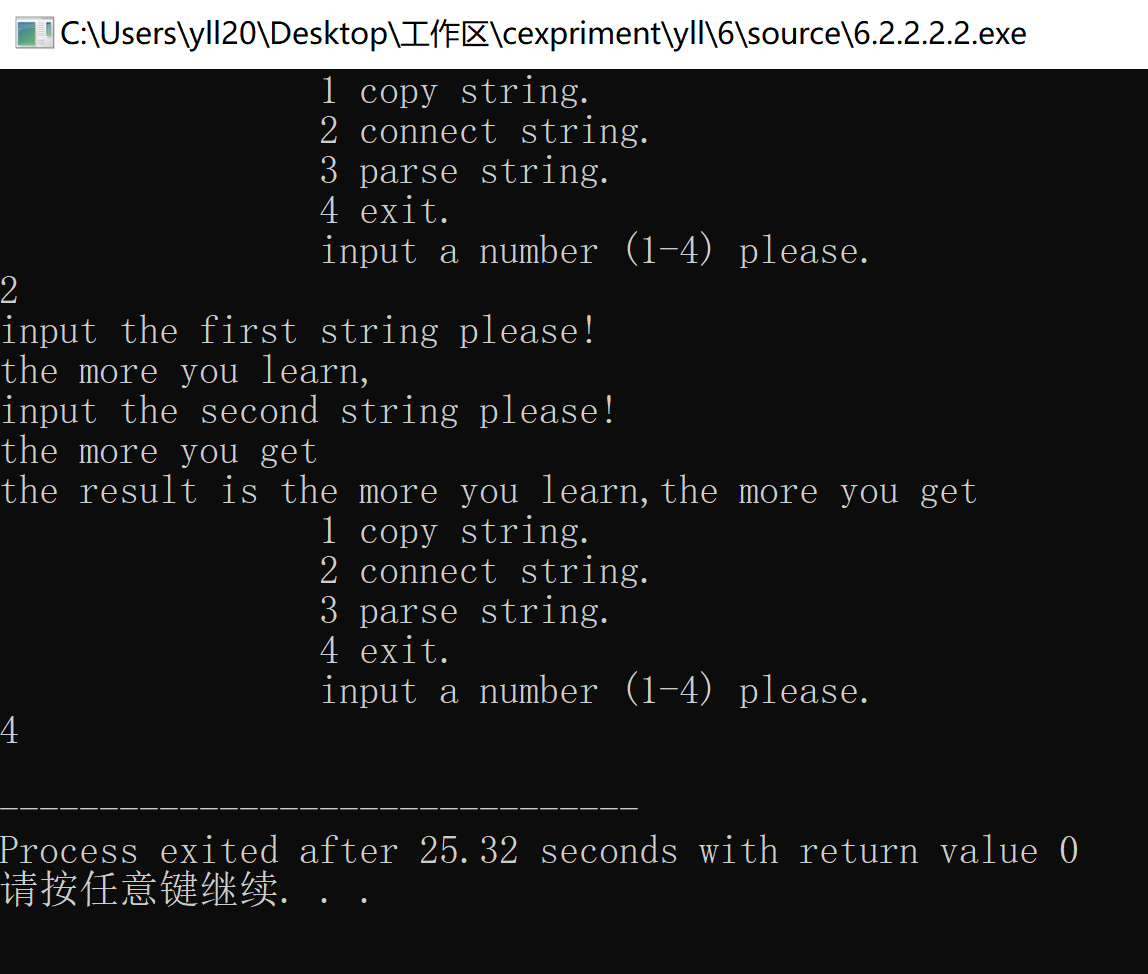


图6-5 程序替换题1的运行结果

**6.2.3 跟踪调试题**

请按下面的要求对源程序进行操作，并回答问题和排除错误。

（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？

（2）排除错误，使程序输出结果为：there is a boat on the lake.

#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);

}

**解答：**

1）修改后程序如下

#include "stdio.h"

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

void main(void)

{

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

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

}

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

{

char \*tmp=s;

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

return (tmp);//返回初始的指针

}

3）测试

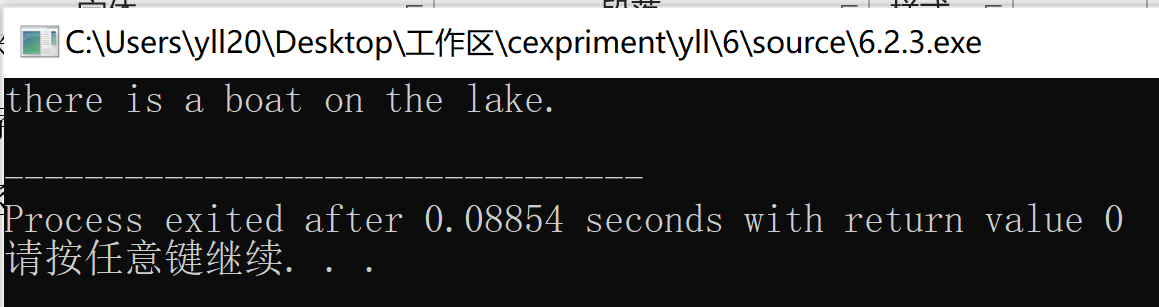


图6-6 程序设计题1的运行结果图

（2）编写一个C程序，要求采用模块化程序设计思想，将相关功能用函数实现，并提供菜单选项。该程序具有以下功能：

①输入n个学生的姓名和C语言课程的成绩。

②将成绩按从高到低的次序排序，姓名同时进行相应调整。

③输出所有学生的姓名和C语言课程的成绩。

1） 算法流程如图5-6所示。

图5-6 程序设计题2的程序流程图

2）源程序清单

#include<stdio.h>

#include<windows.h>

#include <string.h>

char name[100][10]={};

char name\_temp[100][10]={};

int score[100]={0};

int score\_temp[100]={0};

void input();

void sort();

void output();

void search();

void sort2();

int main()

{

static char temp;

system("title 学生成绩管理系统 by yll");

start:

system("Cls");

printf("-----------------------\n");

printf("| 学生成绩管理系统 |\n");

printf("-----------------------\n");

printf("| 1 |输入学生成绩|\n");

printf("-----------------------\n");

printf("| 2 | 按成绩排序 |\n");

printf("-----------------------\n");

printf("| 3 |输出所有成绩|\n");

printf("-----------------------\n");

printf("| 4 | 查找成绩 |\n");

printf("-----------------------\n");

printf("| 5 | 退出系统 |\n");

printf("-----------------------\n");

scanf("%c",&temp);

if (temp=='1')

{

input();

}

if (temp=='2')

{

sort();

}

if (temp=='3')

{

output();

}

if (temp=='4')

{

search();

}

if (temp=='5')

{

return 0;

}

else

{

printf("输入错误！\n");

}

goto start;

return 0;

}

void input()

{

system("Cls");

int i=0;

printf("-----------------------------------\n");

printf("|请这样输入学生成绩： |\n");

printf("|张三 96 |\n|李四 98 |\n");

printf("|以ctrl+Z结束录入 |\n");

printf("-----------------------------------\n");

while(scanf("%s",&name[i][0])!=EOF)

{

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

i++;

}

}

void output()

{

system("Cls");

int i=0;

printf("-----------------\n");

printf("|姓名\t|成绩\t|\n");

printf("-----------------\n");

while(score[i])

{

printf("|%s\t|%d\t|\n",name[i],score[i]);

printf("-----------------\n");

i++;

}

printf("请按任意键返回......\n");

getchar();

getchar();

}

void sort()

{

int len=0;

while(score[len])

{

len++;

}

int i, j;

int v;

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

{

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

{

if(score[i] < score[j])//如前面的比后面的小，则交换。

{

v = score[i];

score[i] = score[j];

score[j] = v;

char temp[10];

strcpy(temp,name[i]);

strcpy(name[i],name[j]);

strcpy(name[j],temp);

}

}

}

output();

}

void sort2()

{

int len=0;

while(score[len])

{

len++;

}

int i, j;

int v;

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

{

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

{

if(score\_temp[i] < score\_temp[j])//如前面的比后面的小，则交换。

{

v = score\_temp[i];

score\_temp[i] = score\_temp[j];

score\_temp[j] = v;

char temp[10];

strcpy(temp,name\_temp[i]);

strcpy(name\_temp[i],name\_temp[j]);

strcpy(name\_temp[j],temp);

}

}

}

}

void search()

{

system("Cls");

printf("-----------------------\n");

printf("| 输入要查找的成绩 |\n");

printf("-----------------------\n");

int target = 0;

scanf("%d",&target);

int len=0;

while(score[len])

{

score\_temp[len] = score[len];

strcpy(name\_temp[len],name[len]);

len++;

}

int left = 0;

int right = len - 1;

int flag=1;

sort2();

while(flag)

{

while(left <= right)

{

int mid = left+(right-left)/2; //防止溢出

if(score\_temp[mid] == target)

{

printf("-----------------\n");

printf("|%s\t|%d\t|\n",name[mid],score[mid]);

printf("-----------------\n");

score\_temp[mid] = -2; //不会真有人考-2分吧

}

else if (score\_temp[mid] < target)

right = mid - 1;

else if (score\_temp[mid] > target)

left = mid + 1;

}

flag = 0;

printf("查找结束......\n");

}

printf("请按任意键返回......\n");

getchar();

getchar();

}

3）测试

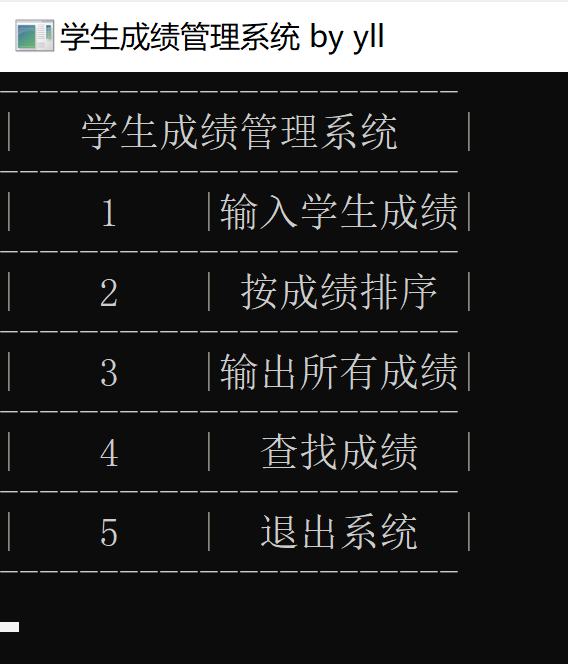


图5-6 程序设计题2的运行结果图1

****

图5-7 程序设计题2的运行结果图2



图5-8 程序设计题2的运行结果图3

**5.2.4 程序设计选做题**

编写并上机调试运行能实现以下功能的函数和程序。

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。

**解答：**

1） 算法流程如图5-9所示。

图5-9 选做题题的程序流程图

2）源程序清单

#include<stdio.h>

void strnins(char a[],char b[],int n);

int main()

{

char a[100],b[100];

int n=0;

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

strnins(a,b,n);

printf("%s",a);

return 0;

}

void strnins(char a[],char b[],int n)

{

int i=n;

int j=0;

while(b[j])

{

a[i] = b[j];

i++;

j++;

}

a[i+1]='\0';

}

3）测试

（a） 测试数据：

ilovechina ilovehust 5

（b） 对应测试数据的运行结果截图

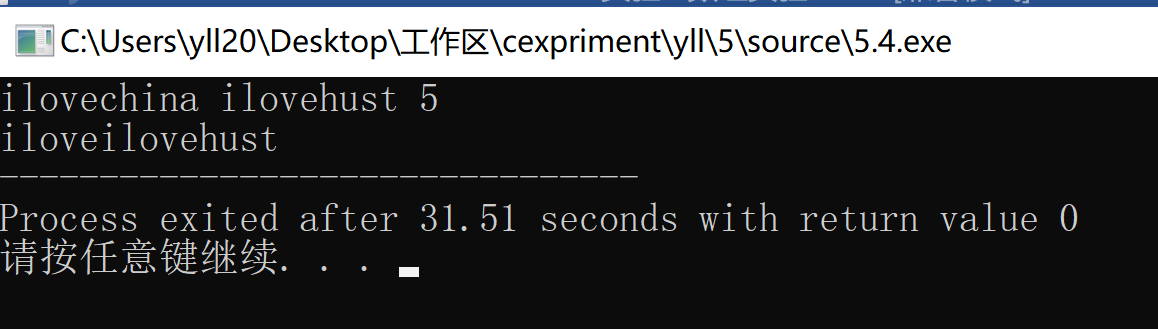


图5-10 选做题的运行结果图

## 5.4 实验小结

实验中体会到了数组在存储数据上的方便性，同时也发现使用数组的时候一定要小心访问非法内存的问题。

参考文献

[1] 曹计昌,卢萍,李开. C语言程序设计,北京： 科学出版社,2013

[2] 李开,卢萍,曹计昌. C语言实验与课程设计, 北京：科学出版社,2011