

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 软件工程2003班**

**学 号： U202010783**

**姓 名： 刘铭宸**

**指导教师： 唐赫**

**报告日期： 2020.12.11**

**软件工程**

**目 录**

[**6 指针实验**](#_Toc404837943) **1**

[6.1 实验目的 1](#_Toc404837921)

[6.2 实验内容 1](#_Toc404837922)

[6.3 实验小结](#_Toc404837923) 24

# 6 指针试验

**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**

**1．程序改错**

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

#include<stdio.h>

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

int main(void)

{

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

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

scanf("%s", s2);

strcopy(s1, s2);

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

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

scanf("%s", s2);

s3 = strcopy(s1, s2);

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

return 0;

}

/\*将字符串s复制给字符串t，并且返回串t的首地址\*/

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

{

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

return (t);

}

**解答：**

（1）错误修改：

1)第5行应给s1,s2,s3分配内存，同时应引入头文件，正确形式为：

#include<stdlib.h>

char\*s1=(char\*)malloc(sizeof(char)\*100),\*s2=(char\*)malloc(sizeof(char)\*100), \*s3=(char\*)malloc(sizeof(char) \* 100);

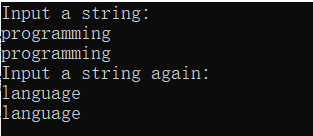
2)在strcopy函数中t指针发生了变化，未能正确返回串t的首地址，应另设一指针指向t的首地址，正确形式为：

char\* p = t;

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

return (t);

（2）错误修改后运行结果：



**6.2.2 源程序完善和修改替换**

（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（字符串长度不超过49）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-i-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]); free(s[i]);}

return 0;

}

**解答：**

程序中使用了malloc函数，故应引入头文件<stdlib.h>；temp作为交换s[j]和s[j+1]的中间变量，数据类型应与s[j]相同；第11行的判断条件应是s[j]的字典序比s[j+1]高，故使用库函数strcmp。

#include<stdio.h>

#include<stdlib.h>

#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 - i - 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\_s(t);

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

strcpy(s[i],t);

}

strsort(s,N);

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

return 0;

}

**测试：**

（a） 测试数据：

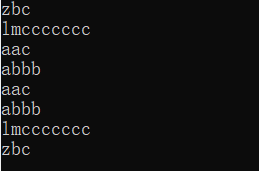
zbc

lmccccccc

aac

abbb

（b） 对应测试数据的运行结果截图



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：**

定义两个二级指针p和q，使p首先指向s中第一个字符串的首地址，q指向p后一位，在每次循环中使p和q后移一位，每次让\*p和\*q交换，即可达到不使用下标引用完成函数功能的效果。

#include<stdio.h>

#include<stdlib.h>

#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;

char\*\* p=s, \*\* q;

int i, j;

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

for (j = 0,p=s,q=s+1; j < size - i - 1; j++,p++,q++)

if (strcmp(\*p,\*q) > 0)

{

temp = \*p;

\*p = \*q;

\*q = temp;

}

}

int main()

{

int i;

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

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

{

gets\_s(t);

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

strcpy(s[i], t);

}

strsort(s, N);

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

return 0;

}

**测试：**

（a） 测试数据：

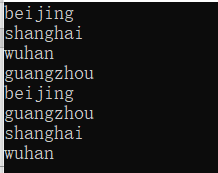
beijing

shanghai

wuhan

guangzhou

（b） 对应测试数据的运行结果截图



（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 = strtok; break;

case 4: 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;

}

**解答：**

p应为一函数指针，根据用户输入的choice的值分别指向strcpy、strcat、strtok函数，在使用时格式为p(a,b)。

# 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);//

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

gets(b);//

result = p(a, b);//

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

}

down:

return 0;

}

**测试：**

（a） 测试数据：

1

hello

world

2

world

helloworld

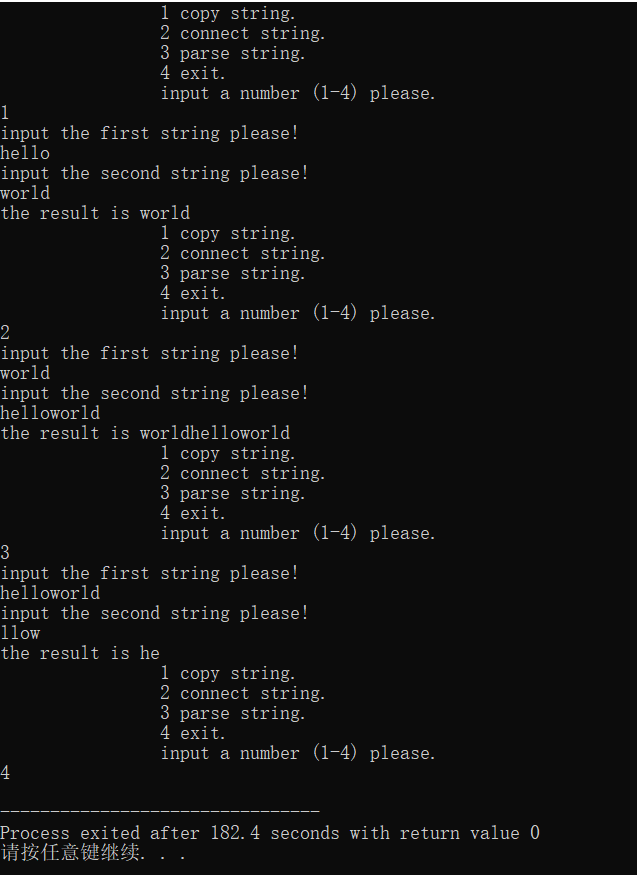
3

helloworld

llo

4

（b） 对应测试数据的运行结果截图



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：**

将p定义为一个函数指针数组，将strcpy、strcat、strtok函数作为元素放入p中，将用户输入的choice值作为数组下标分别调用三个函数，即可避免用switch语句对p进行不同情况的赋值。

# include<stdio.h>

# include<string.h>

int main (void)

{

char\* (\*p[]) (char\* , char\*) = {0, strcpy, strcat, 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)

break;

getchar();

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

gets(a);//

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

gets(b);//

result = p[choice](a, b);//

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

}

down:

return 0;

}

**测试：**

（a） 测试数据：

1

huazhong

zhong hua

2

zhonghua

hua zhong

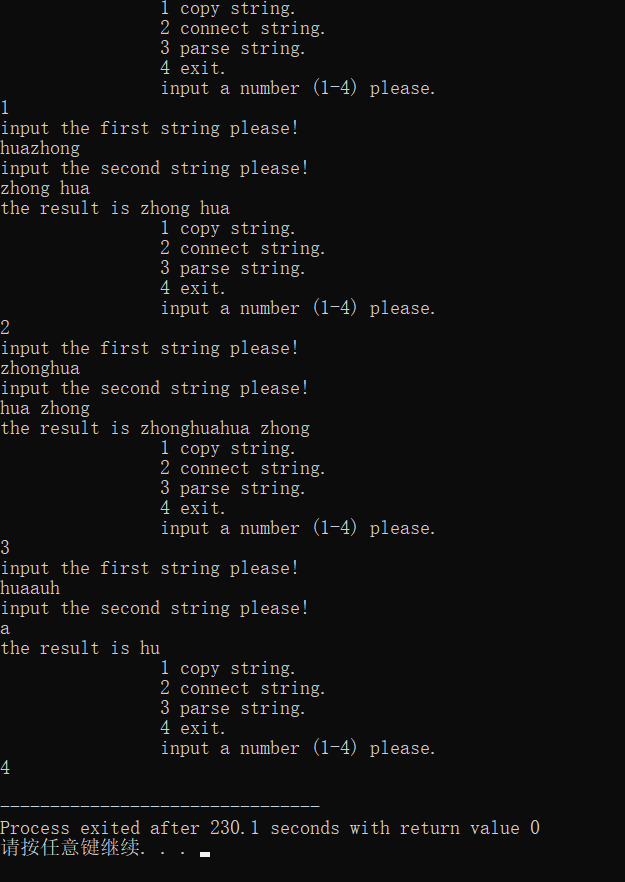
3

huaauh

a

4

（b） 对应测试数据的运行结果截图



**6.2.3 程序跟踪调试**

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。

#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)

{

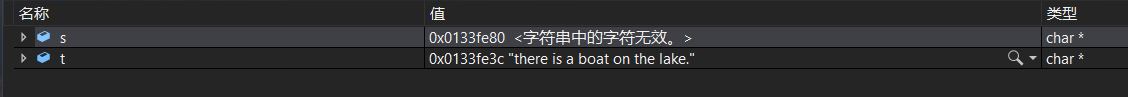
while(\*s++=\*t++)

;

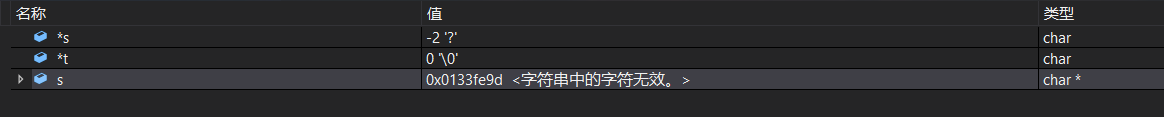
return (s);

}

（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？



如图，进入strcpy时s未被初始化，字符串中的字符无效。



如图，返回main时，s中的字符无效

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

**解答：**

数组a容量过小，“there is a boat on the lake”已经超过了20个字符；strcpy函数中s指针发生了变化，return(s)不能返回首地址，应另设一指针指向s首地址。



**6.2.4 程序设计**

（1）指定main函数的参数

在IDE（比如DevC++）中，选择“**运行**”｜“**参数**”菜单，在 “**传递给主程序的参数**”文本框中输入main函数的参数arg1 arg2 arg3，只输入命令行中文件名后的参数，文件名不作为参数输入，参数间以空格隔开。编写程序在命令行输出这三个参数。（注意不同IDE输入参数的方式不相同，可参考各个IDE的使用手册。）

**解答：**

1）在main函数中设置两个形参：int类型的argc和二维数组args。argc表示

传入参数的个数，argc用来存储传入的参数。

2）源程序清单

#include <stdio.h>

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

for(int i=1;i<=argc;i++){

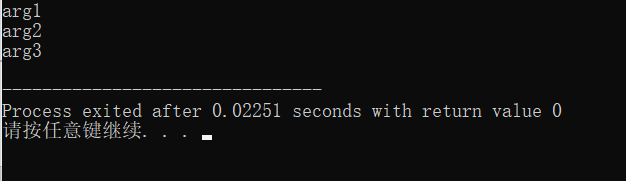
printf("%s\n",args[i]);

}

}

3）测试

对应测试数据的运行结果截图



（2）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。输入一个长整型变量，要求从高字节开始，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，通过指针取出每字节。

样例输入：15

样例输出：0000000F

**解答：**

1. 解题思路：

设置一指针变量，首先指向输入的长整型变量的第一个字节，将每个字节的高4位和低4位取出并以十六进制形式输出。

2）程序清单

#include <stdio.h>

int main(void) {

long long x;

while ((scanf("%lld", &x)) != EOF)

printf("%08X\n", x);

return 0;

}

3）测试

（a） 测试数据：

2147483647

15

4294967295

（b） 对应测试数据的运行结果截图

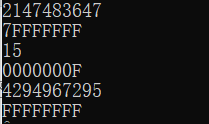


图6-1 编程题2的测试运行结果

（3） 旋转是图像处理的基本操作，编程实现一个将一个图像逆时针旋转90°。提示：计算机中的图像可以用一个矩阵来表示，旋转一个图像就是旋转对应的矩阵。将旋转矩阵的功能定义成函数，通过使用指向数组元素的指针作为参数使该函数能处理任意大小的矩阵。要求在main函数中输入图像矩阵的行数n和列数m，接下来的n行每行输入m个整数，表示输入的图像。输出原始矩阵逆时针旋转90°后的矩阵。

**解答：**

1. 解题思路：

将一个矩阵逆时针旋转90度，就是要把原矩阵中的最后一列作为第一行，直到把第一列作为最后一行。所以应该从最后一列开始输出，每输完一列输出一个换行符。

2）程序清单

#include <stdio.h>

int array[2000][2000];

void out(int, int, int a[][2000]);

int main(void) {

int n, m;

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

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

for (int j = 0; j < m; j++)

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

out(n, m, array);

return 0;

}

void out(int n, int m, int a[][2000]) {

for (int i = m - 1; i >= 0; i--)

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

printf("%d%s", a[j][i],(j==n-1) ? "\n" : " ");

}

3）测试

（a） 测试数据一：

1 1

5

测试数据二：

2 3

1 5 3

3 2 4

测试数据三：

2 3

4 7 8

5 3 2

（b） 对应测试测试用例1的运行结果如图6-2所示。



图6-2 编程题3的测试用例一的运行结果

对应测试测试用例2的运行结果如图6-3所示。

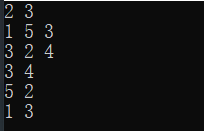


图6-3 编程题3的测试用例二的运行结果

对应测试测试用例3的运行结果如图6-4所示。

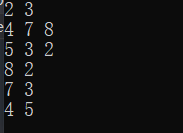


图6-4 编程题3的测试用例三的运行结果

（4）输入n行文本，每行不超过80个字符，用字符指针数组指向键盘输入的n行文本（n不作为输入，可理解为循环输入多行，以CTRL+Z结束），删除每一行中的前置空格（' '）和水平制表符（'\t'）。

要求：将删除一行文本中前置空格和水平制表符的功能定义成函数，在main函数中输出删除前置空格符的各行。（并在最后输出换行符。）

**解答：**

1. 解题思路：

设置一个标志数组，将其初值都设为0，当标志数组序号对应的字符是每一行中的前置空格或者水平制表符时，将其标志位设为1。输出时判断每个字符所对应的标志元素是否为0，是则输出，否则不输出。

2）程序清单

#include <stdio.h>

void Delete(char str[][100], int flag[][100], int n);

int main(void) {

char str[100][100];

int flag[100][100] = { 0 };

int i = 0, j, n = 0;

while ((gets(str[i])) != NULL) {

n++;

i++;

}

Delete(str, flag, n);

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

for (j = 0; str[i][j]; j++) {

if (!flag[i][j])printf("%c", str[i][j]);

}

printf("\n");

}

return 0;

}

void Delete(char str[][100],int flag[][100], int n) {

int i = 0, j = 0, k = 0;

while (k < n) {

j = 0;

while (str[k][j] == ' ') {

flag[k][j] = 1;

j++;

}

k++;

}

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

for (j = 0; str[i][j]; j++)

if (str[i][j] == '\t')flag[i][j] = 1;

}

3）测试

（a） 测试数据一：

123/t45

67/t89

测试数据二：

aab c

bcd/te

测试数据三：

a c f/tl

abcd

（b） 对应测试测试用例1的运行结果如图6-5所示。

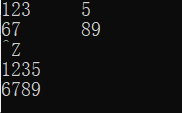


图6-5 编程题4的测试用例一的运行结果

对应测试测试用例2的运行结果如图6-6所示。

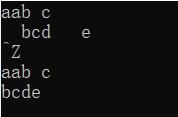


图6-6 编程题4的测试用例二的运行结果

对应测试测试用例3的运行结果如图6-7所示。

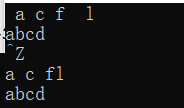


图6-7 编程题4的测试用例三的运行结果

（5） 编写8个任务函数，一个scheduler调度函数和一个execute执行函数。仅在main函数中调用scheduler函数，scheduler函数要求用最快的方式调度执行用户指定的任务函数。

①先设计task0, task1, task2, task3, task4, task5, task6, task7共8个任务函数，每个任务函数的任务就是输出该任务被调用的字符串。例如，第0个任务函数输出“task0 is called!”，第1个任务函数输出“task1 is called!”，以此类推。

②scheduler函数根据键盘输入的数字字符的先后顺序，一次调度选择对应的任务函数。例如，输入：1350并回车，则scheduler函数一次调度选择task1, task3, taks5, task0，然后以函数指针数组和任务个数为参数将调度选择结果传递给execute函数并调用execute函数。

③execute函数根据scheduler函数传递的指针数组和任务个数为参数，按照指定的先后顺序依此调用执行选定的任务函数。

**解答：**

1. 解题思路：

设置一个函数指针数组，每当用户输入数字时，将此数字代表的函数地址作为元素存放在数组中，并将此数组作为参数传进execute函数中，依次按用户输入的顺序，即数组中元素的顺序执行函数。

2）程序清单

#include <stdio.h>

void task0() { printf("task0 is called!"); }

void task1() { printf("task1 is called!"); }

void task2() { printf("task2 is called!"); }

void task3() { printf("task3 is called!"); }

void task4() { printf("task4 is called!"); }

void task5() { printf("task5 is called!"); }

void task6() { printf("task6 is called!"); }

void task7() { printf("task7 is called!"); }

void scheduler();

void execute(void (\*fun[100])(), int n);

int main(void) {

scheduler();

return 0;

}

void scheduler() {

char str[1][100];

int count = 0,count1=0;

void (\*fun[100])();

scanf("%s", str[0]);

do

{

switch (str[0][count])

{

case '0':fun[count1] = task0; break;

case '1':fun[count1] = task1; break;

case '2':fun[count1] = task2; break;

case '3':fun[count1] = task3; break;

case '4':fun[count1] = task4; break;

case '5':fun[count1] = task5; break;

case '6':fun[count1] = task6; break;

case '7':fun[count1] = task7; break;

default:count1--;

}

count++;

count1++;

} while (str[0][count]);

execute(fun, count1);

}

void execute(void (\*fun[100])(), int n) {

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

fun[i]();

if(i!=n-1)printf("\n");

}

}

3）测试

（a） 测试数据一：

13607122

测试数据二：

6735643

测试数据三：

012345678

（b） 对应测试测试用例1的运行结果如图6-8所示。

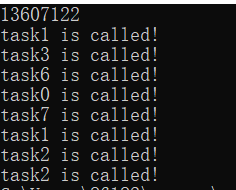


图6-8 编程题5的测试用例一的运行结果

对应测试测试用例2的运行结果如图6-9所示。

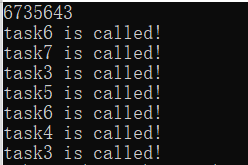


图6-9 编程题5的测试用例二的运行结果

对应测试测试用例3的运行结果如图6-10所示。

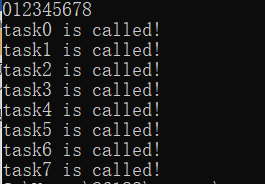


图6-10 编程题5的测试用例三的运行结果

## 6.3 实验小结

主要叙述实验过程中遇到的问题，如何解决的，通过分析、结果问题后的体会。

在源程序完善和修改替换中，我知道了strcpy、strcmp等库函数的用法和作用，掌握了不用数组下标而使用二级指针表示数组和数组元素的方法，了解了函数指针的作用与用法，体会到了用函数指针数组表示多个函数的方便性；在编程题一中，我了解了向main函数传递参数的方法；在编程题5中，我进一步巩固了函数指针和函数指针数组方面的知识。