

# 云南大学数学与统计学实验教学中心

## 《高级语言程序设计》实验报告

课程名称：程序设计和算法语言	学期：2016~2017 学年上学期	成绩：
指导教师：赵越	学生姓名：刘鹏	学生学号：20151910042
实验名称：指针程序设计(2)		
实验编号：No.08	实验日期：2017 年 8 月 20 日	实验学时：2
学院：数学与统计学院	专业：信息与计算科学	年级：2015 级

### 一、实验目的

1. 掌握指针变量的定义，会使用指针变量。
2. 掌握指针与变量、指针与数组、指针与字符串的关系。

### 二、实验环境

Windows10 Pro Workstation 17096;  
Cygwin GCC 编译器。

### 三、实验内容

#### 3.1 知识准备

- 指针就是地址。指针变量是存放另一个变量的地址的变量，不要把指针和指针变量两个概念混淆。
- 定义指针变量后，一定要给指针变量赋初值，使用没有初始指向的指针是非常危险的。
- 指针变量作函数形参，可以接受来自实参的值（地址）。

#### 3.2 元素换位

有三个整数 $x$ ,  $y$ ,  $z$ , 设置三个指针变量 $p_1$ ,  $p_2$ ,  $p_3$ , 分别指向 $x$ ,  $y$ ,  $z$ , 然后通过指针变量使 $x$ ,  $y$ ,  $z$ , 三个变量交换顺序, 即原来 $x$ 的值给 $y$ , 把 $y$ 的值给 $z$ ,  $z$ 的值给 $x$ .  $x$ ,  $y$ ,  $z$ 的原值由键盘输入, 要求输出 $x$ ,  $y$ ,  $z$ 的原值和新值. 读懂下面程序, 并上机运行, 把结果打印出来。

##### 3.2.1 程序代码

```
1  /*
2  * filename: 8.1 exchange.c
3  * property: test
4  */
5
6  #include <stdio.h>
7
8  int main() {
9      int x, y, z, t ;
10     int *p1, *p2, *p3;
```

```

11    printf("x, y, z = ");
12    scanf("%d, %d, %d", &x, &y, &z);
13    p1 = &x;
14    p2 = &y;
15    p3 = &z;
16
17    t = *p3;
18    *p3 = *p2;
19    *p2 = *p1;
20    *p1 = t;
21    printf("x = %d\n y = %d\n z = %d", x, y, z);
22
23    return 0;
24 }

```

请学生考虑第 13~15 行为什么不写成: `*p1 = &x; *p2 = &y; *p3 = &z;`; 而第 17 行~20 行不写成: `t = p3; p3 = p2; p2 = p1; p1 = t`。

答: 写成 `*p1=&x; *p2=&y, *p3=&z;` 是错误的, 地址本身是一个数值, 应该赋给一个变量。而对于指针变量而言, 其变量名不带\*;

程序中的互换是发生在实际变量上的, 只是通过指针变量进行了操作。由于输出的是变量的值, 而且是直接引用, 所以交换指针的值是行不通的, 而且 `t` 没有定义指向, 是个危险地址。

### 3.3

编写一个函数 `sort()`, 使 20 个整数由小到大的顺序排列。在主函数中输出排好序的数。本程序编写函数部分要求要用指针来解决。

#### 3.3.1 程序代码

```

1  /*
2  * filename: 8.2 sort.c
3  * property: exercise
4  */
5
6  #include<stdio.h>
7
8  void sort(int array[], int n) {
9      int *p1, *p2, t;
10     for (p1 = array; p1 < array + (n - 1); p1++) {
11         for (p2 = p1 + 1; p2 < array + n; p2++) {
12             if(*p1 > *p2) {
13                 t = *p1;
14                 *p1 = *p2;
15                 *p2 = t;
16             }
17         }
18     }
19 }
20
21 int main() {

```

```

22     int *p, a[20], i;
23     printf("a[20] = ");
24     for (i = 0; i < 20; i++) {
25         scanf("%d", &a[i]);
26     }
27
28     sort(a, 20);
29
30     for (p=a;p<a+20;p++) {
31         if((p - &a[0]) % 5 == 0) {
32             printf("\n");
33         }
34         printf("%4d", *p);
35     }
36
37     return 0;
38 }

```

### 3.3.2 运行结果

```

~/C_Program_Design
Newton@Newton-PC-2 ~/C_Program_Design
$ gcc 8.2\ sort.c -o 8.2.exe
Newton@Newton-PC-2 ~/C_Program_Design
$ ./8.2.exe
a[20] = 1 -9 8 24 6 38 99 521 203 39 5 1 3 4 8 22 0 18 91 2

-9  0  1  1  2
 3  4  5  6  8
 8 18 22 24 38
39 91 99 203 521
Newton@Newton-PC-2 ~/C_Program_Design
$

```

通过此题，学生可进一步理解指针的定义与赋初值的方法。可以思考一下选择排序与冒泡排序的区别，本质上都是两个循环，都是从左往右进行排序，只不过选择排序是间接通过下标进行记录，冒泡排序是直接进行交换，并且可能进行多次交换。

## 3.4 矩阵转置输出

有一个 3\*4 的矩阵，矩阵元素为：

2	4	6	8
1	3	5	7
10	11	12	13

编写一程序实现矩阵的转置。上机调试并运行。

### 3.4.1 程序代码

```

1  /* filename: 8.3 */
2  #include<stdio.h>
3  void Transport(int (*a)[3],int (*b)[3])
4  {
5      int i,j;

```

```

6     for(i=0;i<3;i++)
7     {
8         for(j=0;j<3;j++)
9         {
10            (*(b+j)+i)=*(a+i+j);
11        }
12    }
13 }
14
15 int main()
16 {
17     int i,j;
18     int a[3][3]={2,4,6},{1,3,5},{10,11,12};
19     int b[3][3];
20     Transport(a,b);
21     for(i=0;i<3;i++)
22     {
23         for(j=0;j<3;j++)
24         {
25             printf("%5d",b[i][j]);
26         }
27         printf("\n");
28     }
29     return 0;
30 }

```

### 3.4.2 运行结果

```

D:\OneDrive\C\图指针程序设计(二)\8.3\bin\Debug\8.3.exe
2   1  10
4   3  11
6   5  12

Process returned 0 (0x0)   execution time : 0.119 s
Press any key to continue.

```

## 3.5

有一个整型二维数组，大小为 $m \times n$ ，找出最大值所在的行和列，以及该最大值。此程序要求用一个函数`max()`实现最大值的寻找，并在`max()`函数中最好使用指针解决， $m$ 和 $n$ 为该函数的形参，数组元素的值在主函数中输入。

这是改进的代码，可以实现任意维数的矩阵（只要不是超级大）的求最大值。

### 3.5.1 程序代码

```

1  /* filename: 8.4 */

```

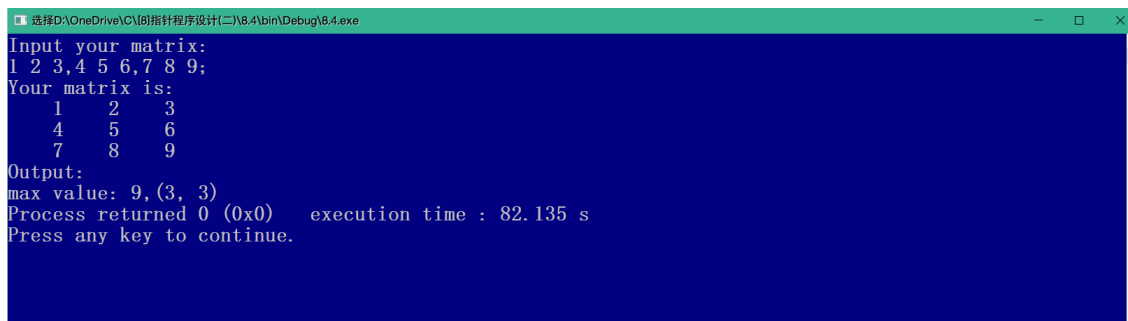
```
2  #include<stdio.h>
3  int rn,cn,r,c,maxval;
4
5  int a[20][20];
6  void input()
7  {
8      int i,j;char m;
9      rn=0;
10     cn=1;
11     printf("Input your matrix:\n");
12     for(i=0;i<20;i++)
13     {
14         cn=1;
15         for(j=0;j<20;j++)
16         {
17             scanf("%d",&a[i][j]);
18             scanf("%c",&m);
19             if(m==' ' || m==';')
20             {
21                 rn++;
22                 break;
23             }
24             else
25             {
26                 cn++;
27             }
28         }
29         if(m==';')
30         {
31             break;
32         }
33     }
34     printf("Your matrix is:\n");
35     for(i=0;i<rn;i++)
36     {
37         for(j=0;j<cn;j++)
38         {
39             printf("%5d",a[i][j]);
40         }
41         printf("\n");
42     }
43 }
44
45 void max()
46 {
47     int i,j;
48     maxval=a[0][0];
49     for(i=0;i<rn;i++)
50     {
```

```

51     for(j=0;j<cn;j++)
52     {
53         if(maxval<*(a+i+j))
54         {
55             maxval=*(a+i+j);
56             r=i+1;
57             c=j+1;
58         }
59     }
60 }
61 }
62
63 int main()
64 {
65     input();
66     max();
67     printf("Output:\nmax value: %d,(%d, %d)",maxval,r,c);
68     return 0;
69 }

```

### 3.5.2 运行结果



The screenshot shows a Windows command prompt window titled "选择D:\OneDrive\...\8.4\bin\Debug\8.4.exe". The program prompts the user to "Input your matrix:" and the user enters "1 2 3,4 5 6,7 8 9;". The program then displays "Your matrix is:" followed by a 3x3 matrix:
 

1	2	3
4	5	6
7	8	9

 The output shows "Output:" followed by "max value: 9,(3, 3)". At the bottom, it says "Process returned 0 (0x0) execution time : 82.135 s" and "Press any key to continue.".

## 3.6

利用指针型参数，写一个函数， $f(x, y)$ 交换 $x$ 和 $y$ 的值。

## 3.7

用指针实现下列功能：从键盘输入一字符串,求此串的长度并输出。

## 3.8 \*

在主函数中输入 10 个等长的字符串。用另一个函数对它们排序，然后输出这 10 个已排好序的字符串。要求用指针解决问题，编好程序上机运行。

## 3.9 程序排错

1. 下列语句中有哪些上不正确的？

(1) `int *p=&i,i,j;`

必须先定义变量，之后将定义的变量的地址存入指针变量，这里的  $i$  定义应该在  $*p$  之前

```
(2) float x,y;
    int *p;
    p=&x;
```

指针基类型与指针指向的变量类型不一致。

```
(3) int i,j;
    i=10;j=20;
    int *p=&i,*q=&j;
```

地址加减的生成值没有变量存放。

```
(4) int i,j,*p[3]={1,2,3},*q=&i+2;
```

指针数组是不能这样赋值的。

```
(5) int m,n,*p,a[10];p=&a[10];
```

a[10]不是一个数组元素，取地址没有意义。

```
(6) int *p=128;
```

对指针进行赋值没有意义。

```
(7) int a=10,*p=a;
```

应该是&a。

```
(8) fun()
    {
        int i,j;
        static int *pi=&i;
    }
```

不会。

```
(9) int i=10,*pi;*pi=i;
```

赋值形似不对。

```
(10) char *ps,s[20]; s="good"; ps=s;
```

s 是数组名字，是静态的，不能被赋值。

### 3.10

先想想，再上机试试，下面加程序输出什么结果。为什么？先看下面程序两种处理方式的差别，再上机验证一下你的判断。

#### 3.10.1 程序代码

```
1  #include<stdio.h>
2  int main()
3  {
4      int i,a[10],b[10],*p=b,*q=&a[0];
5      for (i=0;i<10;i++)
6      {
7          scanf("%d",&a[i]);
8      }
9      for (i=0;i<10;i++,q++)
10     {
11         p[i]=*q;
12     }
13     for (i=0;i<10;i++)
14     {
```

```

15     printf("%d",a[i]);printf("\n");
16 }
17 for(i=0;i<10;i++)
18 {
19     printf("%d",b[i]);
20 }
21 printf("\n");
22 for(i=0;i<10;i++,p++);
23 {
24     printf("%7d",*p);
25 }
26 printf("\n");
27 return 0;
28 }

```

## 四、实验过程

### 4.1 1 题

```

1  /* filename: 8.5 */
2  #include <stdio.h>
3  void swap(int *a,int *b)
4  {
5      int tmp;
6      tmp=*a,*a=*b,*b=tmp;
7  }
8
9  int main()
10 {
11     int a=6,b=8;
12     int *p1=&a,*p2=&b;
13     swap(p1,p2);
14     printf("%d,%d",a,b);
15     return 0;
16 }

```

#### 4.1.1 运行结果

```

D:\OneDrive\VC\08指针程序设计(二)\8.5\bin\Debug\8.5.exe
8,6
Process returned 0 (0x0)   execution time : 0.033 s
Press any key to continue.

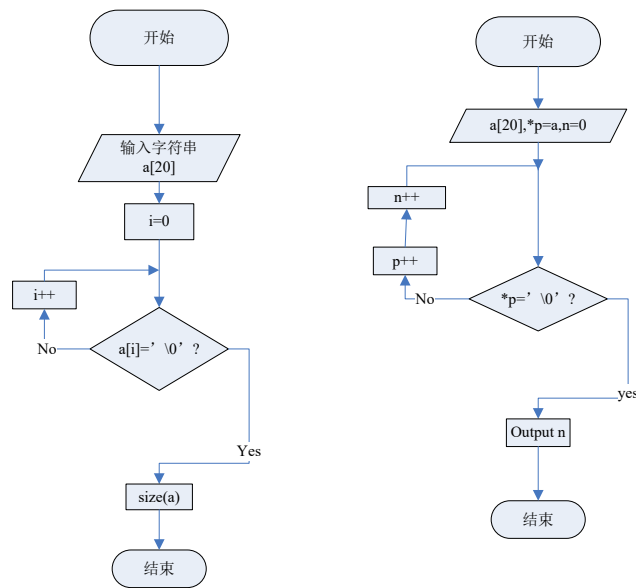
```

### 4.2 2 题:

这个例子考察函数的用法，输入字符串是可以用 `scanf` 与 `gets` 的，不同的是，前者在定义的时候需要声明格式，`%s` 只能输入没有空格的字符串，而 `%[^\n]` 就可以，一直输入（只要不超界），直到遇到换行符才停



止。gets 的用法与 scanf 的 %[^\n] 基本一致。

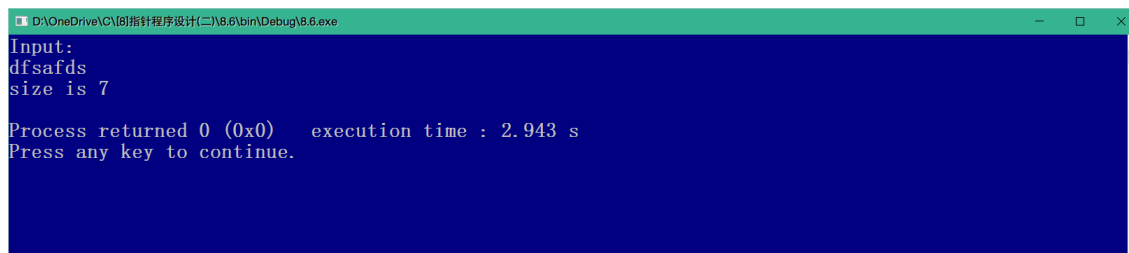


```

1  /* filename: 8.6 */
2  #include<stdio.h>
3  void PrintSize(char a[])
4  {
5      char *p;
6      int n=0;
7      for(p=a;*p!='\0';p++)
8      {
9          n++;
10     }
11     printf("size is %d\n",n);
12 }
13
14 int main()
15 {
16     char a[30];
17     printf("Input:\n");
18     gets(a);
19     /*scanf("%[^\n]",a);*/
20     PrintSize(a);
21     return 0;
22 }

```

### 4.2.1 运行结果



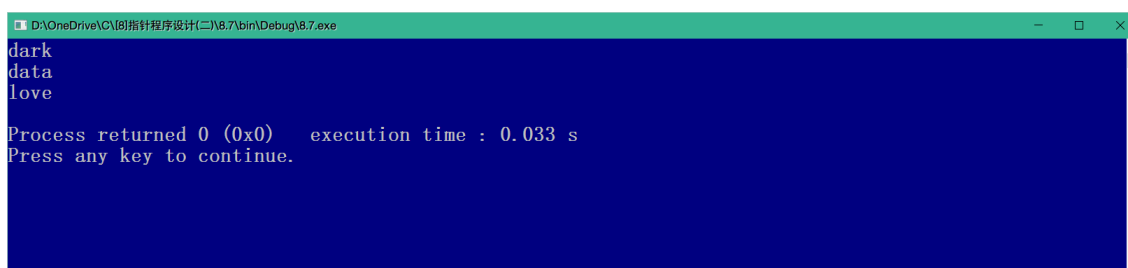
```
D:\OneDrive\C\B\指针程序设计(二)\8.6\bin\Debug\8.6.exe
Input:
dfsafds
size is 7

Process returned 0 (0x0)   execution time : 2.943 s
Press any key to continue.
```

### 4.3 3 题:

```
1  /* filename: 8.7 */
2  #include<stdio.h>
3  #include<string.h>
4  void Sort(char *p[])
5  {
6      int i,j;
7      char *m=NULL;
8      for(i=0;i<2;i++)
9      {
10         for(j=i+1;j<3;j++)
11         {
12             if(strcmp(p[i],p[j])>0)
13             {
14                 m=p[i];
15                 p[i]=p[j];
16                 p[j]=m;
17             }
18         }
19     }
20 }
21
22 void Print(char *name[])
23 {
24     int i;
25     for(i=0;i<3;i++)
26     {
27         printf("%s\n",name[i]);
28     }
29 }
30
31 int main()
32 {
33     char *name[]={"data","love","dark"};
34     Sort(name);
35     Print(name);
36     return 0;
37 }
```

### 4.3.1 运行结果



```
D:\OneDrive\C\...\bin\Debug\8.7.exe
dark
data
love
Process returned 0 (0x0) execution time : 0.033 s
Press any key to continue.
```

## 五、实验总结

由于指针可以对地址进行存储，所以在函数较多的情况下，使用指针可以降低程序运行的内存占用。也由于这个，可以对非全局变量进行函数操作，特别是实参与形参是值传递的情况。

随着时间的推移，我的编译器选择历经轮转，从最初古老的 TC2.0，到 Code::Blocks 集成开发环境，再到 Visual Studio 2017，最终还是回到了 GNU 平台上来，使用开源的一套库进行实验。在此期间，云南大学也从一个普通的 211 大学跻身双一流大学行列，高级语言程序设计这门课程是否也该升级一下？TC2.0 这个编译器，界面十分古朴，在几十年前绝对算是一流的软件，但是现在，确实落后了，这主要是因为它无法编译在 64 位系统下运行的程序。但是 TC2.0 有很多的优势，比如完全可视化的编译过程，不会生成很多附加文件，这一点 Visual Studio 就太过专业化。经过对《UNIX 环境高级编程》[1]这本书的学习，还有诸如 *Harley Hahn's Guide to Unix and Linux*[2]这本书的阅读，我觉得基于 Shell 的 UNIX 环境似乎是最适合新手学习的。

本次实验，集中主要精力，在以前版本的基础上，对文档结构进行了重整，看起来自然了很多，目录也规范了很多。有关编程的规范性问题，参考林锐高质量 C/C++编程指南的第一版[3]。

## 六、参考文献

1. Stevens, W.R. and S.A. Rago, *UNIX 环境高级编程*. 2nd ed. 2005, 北京: 人民邮电出版社.
2. Hahn, H., *Harley Hahn's Guide to Unix and Linux*. 2009, New York: McGraw-Hill.
3. 林锐, *高质量 C++/C 编程指南*. 1.0 ed. 2001.

## 七、教师评语