实验八　指针

【实验目的】

1．掌握变量地址与指针变量的概念，掌握指针的定义与引用方法。

2．掌握指针与一维数组的关系，熟练掌握用指针引用数组元素的方法。

3．掌握指针与字符串的关系，熟练掌握用指针引用字符串的方法。

4．了解指针与二维数组的关系，掌握用指针引用二维数组元素的方法。

5．了解多级指针的概念及其使用方法。

【实验内容】

1．通过输出变量的地址，掌握不同类型变量在内存中所占的字节数。

```
#include<windows.h>
#include <stdio.h>
void main()
{ int a,b; float x,y; char m,n;
  a=3;b=7;
  x=2.7; y=2.5;
  m='h';n='d';
  printf("&a=%d, &b=%d, sizeof(a)=%d\n",&a,&b,sizeof(a));
  printf("&x=%d, &y=%d ,sizeof(x)=%d \n",&x,&y,sizeof(x) );
  printf("&m=%d, &n=%d,sizeof(m)=%d \n",&m,&n ,sizeof(m));
  system("pause");
}
```

运行该程序，观察同种类型两个变量的地址之差是否与用 sizeof 所求本类型变量所占字节数相同。

2．通过指针变量的加减运算，理解指针变量移动的字节数。

```
#include<windows.h>
#include <stdio.h>
int main()
{
int a=4,b=7;
    int *pa=&a,*pb=&b,*p1,*p2;
    printf("*pa=%d,*pb=%d\n",*pa,*pb); /* *pa 和*pb 应是 a 和 b 的值*/
    pa=pa+1;pb=pb-1;
    printf("*pa=%d,*pb=%d\n",*pa,*pb); /* *pa 和*pb 已不是 a 和 b 的值了*/
    pa=&a;
    p1=pa+1;p2=pa+2;
        printf("pa=%d,p1=%d,sizeof(p1-pa)=%d\n",pa,p1,sizeof(p1-pa));
// pa 为变量 a 的地址，p1 是 pa+1 的值，
//sizeof(p1-pa) 是 p1 和 pa 之间相差的字节数
    printf("p1=%d,p2=%d,sizeof(p2-p1)=%d\n",pa,p1,p2,sizeof(p2-p1));
              /* p2 是 pa+2 的值，sizeof(p2-p1) 是 p1 和 p2 之间相差的字节数 */
system("pause");
}
```

运行该程序，观察运行结果。验证：

（1）当语句“pa=pa+1;pb=pb-1;”执行过后，*pa 和*pb 已不是 a 和 b 的值了，即 pa 和 pb 已不指向 a 和 b 了。

（2）pa+1 的值正好是变量 a 的地址与 sizeof(p1−pa)的和，sizeof(p2−p1)表明的字节数正是该类型变量存储时在内存中所占的字节数。

（3）应掌握指针变量加 1，指针实际上移动的量是该类型变量存储时在内存中所占的字节数。

3．数值传递与地址传递的比较应用。

```
#include<windows.h>
#include <stdio.h>
void f1(int x,int *y)
{
printf("最初 f1 中 x, *y 的值: %d,%d\n",x,*y);
x=x+10;
*y=*y+10;
printf("变化后 f1 中 x, *y 的值: %d,%d\n",x,*y);
}
int main()
{
int a=10,b=10;
printf("原始 a, b 的值: %d,%d\n",a,b);
f1(a,&b);
printf("调用函数 f1 后 a, b 的值: %d,%d\n",a,b);
system("pause");
}
```

运行该程序，观察运行结果。

说明：由于调用函数 f1(a,&b)中的参数传递 a 为传数值，&b 是传地址，因此 f1 中 x 的值由 10 变为 20 并不影响 a 的数值，因为 a 和 b 是两个不同的整型变量。而 f1 中*y 的值由 10 变为 20 影响了 b 的数值，因为 y 中存储的是 b 的地址，*y 代表的就是 b，*y 的值就是 b 的值，*y 变为 20，b 就变为 20。

4．利用指针输出数组的元素值。

参考程序如下：

```
#include<windows.h>
#include <stdio.h>
int main()
{
int a[10]={1,2,3,4,5,6,7,8,9,10},*n;
    for(n=a;n<a+10;n++)
        printf("%d,",*n);
    printf("\n");
system("pause");

}
```

由上题可知，处理数组的问题，用指针比较方便。

5．用指针作为函数的参数，由键盘输入 3 个数，按由小到大的顺序排序并显示出来。在横线处填写代码，使程序完整。

```
#include <stdio.h>
```

```
void sort(int *p1,int *p2,int *p3)
{
    ____________
    if(________)
        {t=*p1;*p1=*p2;*p2=t;}
    if(*p2>*p3)
        {t=*p2;*p2=*p3;*p3=t;}
    if(*p1>*p2)
        {t=*p1;*p1=*p2;*p2=t;}
}
main()
{
    int a,b,c;
    printf("please input three numbers:\n");
    scanf("%d%d%d",&a,&b,&c);
    ___________
    printf("a=%d,b=%d,c=%d\n",a,b,c);
}
```

思考题：要求用教材上例 8.13 中的 swap 函数来做本题，将如何实现。

6. 用函数与指针来完成，对由 10 个数组成的一维数组的元素按由大到小的顺序排序，并在主调函数中输出。

参考程序如下：

```
#include <stdio.h>
void sort(int a[],int n)
{ int i,j,t;
  for(i=0;i<n-1;i++)
     for(j=i+1;j<n;j++)
       if(a[i]<a[j])
       {t=a[i];a[i]=a[j];a[j]=t;}
}
main()
{ int aa[10]={1,2,3,4,5,6,7,8,9,10},i,n=10;
  sort(aa,n);
  for(i=0;i<10;i++) printf("%d,",aa[i]);
  printf("\n");
}
```

将语句 `void sort(int a[],int n)` 改为 `void sort(int *a,int n)`，然后再运行程序，观察结果。

思考题：如何改变主函数，来实现对由 10 个数组成的一维数组中任意个连续的元素按由大到小的顺序排序，并在主调函数中输出。

主函数参考程序如下：

```
main()
{ int aa[10]={1,2,3,4,5,6,7,8,9,10},i,m,n;
  scanf("%d%d",&m,&n);  /* m 为从第几个数开始排序，n 为排几个数*/
  sort(&aa[m-1],n);
```

```
    for(i=0;i<10;i++) printf("%d,",aa[i]);
    printf("\n");
}
```

7. 用指针来处理字符串的应用。程序如下：

```
#include <stdio.h>
#include <string.h>
main()
{
    char *a="my name is zhongguo",b[20];
    printf("%s\n",a);
    printf("%c,%c,%c\n",*a,*(a+1),*(a+4));
    printf("%c,%c,%c\n",a[0],a[1],a[4]);
    b[3]=*(a+1);
    printf("%c,%c\n",b[3],*(b+3));
    strcpy(b,a);
    printf("%s\n%s\n",a,b);
}
```

观察程序的运行结果，输出*(a+i)和 a[i]是相同，都代表是 a 字符串的第 i+1 个字符。对于赋值，字符赋值可以直接进行，如，b[3]=*(a+1)；字符串赋值必须通过函数进行，如，strcpy(b,a)。

8. 用函数与指针来完成，判断一个数是否为回文数。如，1234321、123321 为回文数。在横线处填写代码，使程序完整。

```
#include <stdio.h>
#include <string.h>
void f(char *s)
{
    int n;
char *p=s,*m;
    n=strlen(s);m=s+n-1;
    for(;p!=s+n/2;p++,m--)
        if(________)break;
    if(p==s+n/2)
        printf("_______");
    else
        printf("_______");
}
main()
{
char a[30];
gets(a);
f(a);
}
```

将主函数改为如下程序并运行，观察结果：

```
main()
{
```

```
    char *a="12321";
    f(a);
}
```

将主函数改为如下程序并运行，观察是否出现问题：

```
main()
{
    char *a;
    gets(a);
    f(a);
}
```

9. 用函数与指针来完成：把 b 字符串连接到 a 字符串的后面，并返回 a 中新字符串的长度。

参考程序如下：

```
#include <stdio.h>
int strlen(char a[], char b[])
{ int num=0,n=0;
  while(*(a+num)!= '\0') num++;
  while(b[n])
  {*(a+num)=b[n]; num++; n++;}
  return(num);
}
main()
{ char s1[]="abcd";
  char *s2="efghi";
  int len;
  len=strlen(s1,s2); printf("%d\n",len);
}
```

思考题：若将“char *s2="efghi";”改为“char s2[]="efghi";”可不可以？若将“char s1[]="abcd";”改为“char *s1="abcd";”可不可以？

10. 指针、指针数组与多级指针的应用。

（1）用简单指针变量引用二维数组元素

```
#include <stdio.h>
main()
{
   int a[2][4]={{1,2,3,4},{5,6,7,8}};
   int i,*p=a[0];
   for(i=1;p<a[0]+8;p++,i++)
     {
         printf("%4d",*p);
         if(i%4==0)
             printf("\n");
     }
}
```

思考：假如将语句*p=a[0]换成*p=a 是否可以？

说明：可以把二维数组 a 看成由两个一维数组 a[0]和 a[1]组成，a[0]代表了一维数组 a[0]

的首地址，也就是 a[0][0]的地址。而 a 代表的是二维数组 a 的首地址，即一维数组 a[0]的地址。

（2）用指向二维数组一整行的指针变量引用二维数组元素

```
#include <stdio.h>
main()
{
   int a[4][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
   int(*p)[4];   //p 为指向二维数组一整行的指针
   int i;
   for(p=a;p<a+4;p++)
       for(i=0;i<4;i++)
         {
             printf("%6d",*(*p+i));
             if((i+1)%4==0)
                 printf("\n");
         }
}
```

分析：由于 p 是指向一行的指针，它的加 1 操作是以行为单位的，因此外循环以 p 为循环变量，若 p++，则 p 向下移动一行。内循环是输出一行内的 4 个元素，由于 p 是二级指针且初值为 a，因此取一维数组的第 1 个元素的地址（一级指针）为*p，第 2 个元素的地址为*p+1，第 i 个元素的地址为*p+i。由地址取存储内容为在地址前加一个*即可，如*(*p+i)。

（3）用指针数组引用二维数组元素

```
#include <stdio.h>
main()
{
   int a[4][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
   int *p[4]={a[0],a[1],a[2],a[3]};
   int i,j;
   for(i=0;i<4;i++)
       for(j=0;j<4;j++)
         {
             printf("%6d",*(p[i]+j));
             if((j+1)%4==0)
                 printf("\n");
         }
}
```

分析：p 数组有 4 个元素，分别为 p[0]～ p[3]，取值依次为 a[0]、a[1]、a[2]、a[3]，分别是 4 个一维数组的首地址（第 1 个元素的地址），第 i 行的首地址为 p[i]，第 i 行第 j 列元素的地址为 p[i]+j。由地址取存储内容为*(p[i]+j)。

（4）用指针数组处理字符串数组，输出 5 个城市名。

```
#include <stdio.h>
main()
{
  char *s[5]={"Dalian","Beijing","Shanghai","Tianjin","Chongqing"};
  int j;
  for(j=0;j<5;j++)
```

```
    printf("%s ",s[j]);
}
```

说明：指针数组 s 中存储的是指向每个城市名字符串的首地址。

（5）用二级指针引用二维数组的元素。

```
#include <stdio.h>
main()
{  int a[4][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
   int *p[4]={a[0],a[1],a[2],a[3]};
   int j,**q;
   for(q=p;q<p+4;q++)
   for(j=0;j<4;j++)
   {  printf("%6d",*(*q+j));
      if((j+1)%4==0)
         printf("\n");
   }
}
```

分析：把指针数组 p 的首地址赋给 q，q 首先指向 p 数组的第一个元素 p[0]，q+i 指向于 p 数组的第 i 个元素 p[i]，*q 是 q 所指向数组 p 某个元素 p[i]的值，即一维数组 a[i]的首地址，而*q+j 是一维数组 a[i]第 j 列元素的地址，*(*q+j) 是一维数组 a[i]第 j 列元素的数值。

（6）用指向指针的指针输出 5 个城市名。

```
#include <stdio.h>
main()
{    char *s[5]={"Dalian","Beijing","Shanghai","Tianjin","Chongqing"};
     int j;
     char **p;
     for(j=0,p=s;j<5;j++,p++)
         printf("%s ",*p);
}
```

分析：第一次循环，把指针数组 s 的首地址赋给 p，p 首先指向 s 数组的第一个字符串 s[0]，*p 是 s[0]的值，即第一个字符串。第二次循环时 p+1，此时 p 指向 s 数组的第二个字符串 s[1]，*p 是 s[1]的值，即第二个字符串，其他依此类推。

11、用函数与指针编程实现以下功能：

1．编一函数，能够返回一维数组中最小值所在的下标值，数组由主调函数传入。

2．编一函数，能够返回字符串的长度，字符串由主调函数传入。

3．编一函数，能够在 N 行 M 列的二维数组中，选出一个最大值作为函数值返回，并通过形参传回此最大值所在的行下标。

4．编一函数，能够将一个数插入一个有序的一维数组中，插入后数组依然有序，数组及插入的数由主调函数传入。

5．编一函数，把 b 字符串连接到 a 字符串的后面，并返回 a 中新字符串的长度。

6．编一函数，通过调用函数返回一维数组中的最大值。

7．编一函数，对由 10 个数组成的一维数组中任意个连续的元素按由大到小的顺序排序，并

在主调函数中输出。

实验八实验内容 11 编程题参考答案

(1)
```
int fun(int *a, int n)
{
    int i,j=0,p;
    p=j;
    for(i=j;i<n;i++)
        if(a[i]<a[p]) p=i;
    return(p);
}
main()
{
    int s[10]={4,5,2,7,1,6,8,3,9,12};
    printf("数组的最小值下标=%d\n",fun(s,10));
}
```

(2)
```
#include <stdio.h>
int mystrlen(char *str)
{
    int i;
    for(i=0; *(str+i)!='\0';i++);
    return i;
}
main()
{
    char *s="djkflfkl";
    int length;
    length=mystrlen(s);
    printf("length=%d\n",length);
}
```

(3)

```
#include <stdio.h>
#define N 3
#define M 3
select(int a[N][M],int *n)
{ int i,j,row=1,colum=1;
   for(i=0;i<N;i++)
      for(j=0;j<M;j++)
         if(a[i][j]>a[row][colum]){row=i;colum=j;}
   *n= row;
   return a[row][colum] ;
}
main()
{ int a[N][M]={9,11,23,6,1,15,9,17,20},max,n;
   max=select(a,&n);
   printf("max=%d,line=%d\n",max,n);
}
```

(4)

```
#include <stdio.h>
#include <string.h>
#define M 10
void f(int a[], int n)
{ int *p,*q,i;
  a[M]=n;
  for(p=a,i=0;i<=M;i++)
    if(n<=*(p+i))
    {
        p=p+i;break;
    }
  for(q=a+M-1;q>=p;q--)
    *(q+1)=*q;
  *p=n;
}
main()
{ int aa[M+1]={2,6,9,12,17,19,22,34,45,56};
  int n,i;
  printf("input number n:\n");
```

```
    scanf("%d",&n);
    f(aa,n);
    for(i=0;i<M+1;i++)printf("%d,",aa[i]);
}
```

(5)

```
#include <stdio.h>
int strlen(char a[], char b[])
{ int num=0,n=0;
  while(*(a+num)!= '\0') num++;
  while(b[n])
  {*(a+num)=b[n]; num++; n++;}
  return(num);
}
main()
{ char s1[20]="abcd";
  char *s2="efghi";
  int len;
  len=strlen(s1,s2); printf("%d\n",len);
}
```

(6)

```
#include <stdio.h>
int func(int *a,int n)
{
    int i,sum=a[0],max;
    max=a[0];
    for (i=1;i<n;i++)
        if (a[i]>max)max=a[i];
    return(max);
}
void main()
{   int i,b[10],max;
    for (i=0;i<10;i++)
        scanf("%d",&b[i]);
    max=func(b,10);          /*调用外部函数 func */
    printf("max=%d",max);
}
```

(7)

```
#include <stdio.h>
void sort(int a[],int m,int n)
{ int i,j,t;
```

```
    for(i=m;i<m+n-1;i++)
        for(j=i+1;j<m+n;j++)
            if(a[i]<a[j])
                {t=a[i];a[i]=a[j];a[j]=t;}
}
main()
{
    int aa[10]={1,2,3,4,5,6,7,8,9,10},i,m,n;
    printf("请输入 m 和 n，m 为从第几个数开始排序(0<m<9)，n 为排几个数(n<10-m)\n");
    scanf("%d%d",&m,&n);
    sort(aa,m-1,n);
    for(i=0;i<10;i++)
        printf("%d,",aa[i]);
    printf("\n");
}
```