Experiment 9 Pointer (I)

1. Purpose of the experiment

(1) Master the concepts of pointer and indirect access, and be able to define and use pointer variables.

(2) Be able to use arrays and pointers and pointer variables pointing to arrays correctly.

(3) Can correctly use the pointer to a string and the pointer variable to a string.

2. Experiment content

Program and run the following programs (all require pointer processing).

(1) Input 3 integers and output them in the order from small to large, and then change the program to: input 3 strings and output them in the order from small to large

(a) First, write a program to process the input of three integers and output them in order from small to large. Run this program to analyze the results.

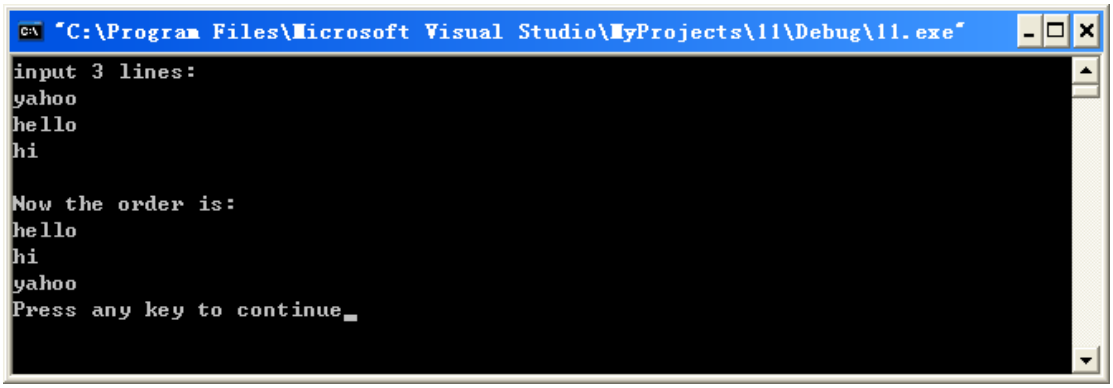
(b) Change the program to be able to input 3 strings and output them in order from small to large. Run this program to analyze the results.

Code:

```
#include <stdio.h>
int main()
{
    void swap(int *p1,int *p2);
    int a,b,c;
    int *p1,*p2,*p3;
    printf("input three integer:\n");
    scanf("%d%d%d",&a,&b,&c);
    p1=&a;
    p2=&b;
    p3=&c;
    if(a>b) swap(p1,p2);
    if(a>c) swap(p1,p3);
    if(b>c) swap(p2,p3);
    printf("\nThe number of outputs is:\n%d %d %d\n",a,b,c);
    return 0;
}
void swap(int *p1,int *p2)
{
    int p;
    {p=*p1;*p1=*p2;*p2=p;}
}
```

(b)Code:

```
#include <stdio.h>
#include <string.h>
int main()
{
    void swap(char *,char *);
    char s1[20],s2[20],s3[20];
    printf("input 3 lines:\n");
    gets(s1);
    gets(s2);
    gets(s3);
    if(strcmp(s1,s2)>0) swap(s1,s2);
    if(strcmp(s1,s3)>0) swap(s1,s3);
    if(strcmp(s2,s3)>0) swap(s2,s3);
    printf("\nNow the order is:\n");
    printf("%s\n%s\n%s\n",s1,s2,s3);
    return 0;
}
void swap(char *a,char *b)
{
    char p[20];
    strcpy(p,a);strcpy(a,b);strcpy(b,p);
}
```



(2)Comparing the above two programs, what is the difference between analyzing and processing integers and processing strings? For example:

(a) How to get a pointer to an integer (or string).

Solution: get integer pointer: declare a pointer variable * p, and take the address to make p=* int

Get string pointer: the string str itself is a pointer, pointing to the address of the first element of the string.

(b) How to compare the size of two integers (or strings).

Solution: Compare integers directly with (>,<,=) and other symbols.

Compare strings, including<string. h>, and call strcmp function.

(c) How to exchange two integers (or strings).

Solution: exchange integers: declare an intermediate variable t so that {t=a; a=b; b=t} can exchange a, b.

Exchange string: it needs to contain<string. h>, call strcpy function, and declare a character group as an intermediate variable.

For example:

char p[20];

strcpy(p,p1);strcpy(p1,p2);strcpy(p2,p);

(3)Transpose a 3x3 shaped two-dimensional array and use it as a function

In the main function, use the scanf function to enter the following array elements:

1	3	5
7	9	11
13	15	19

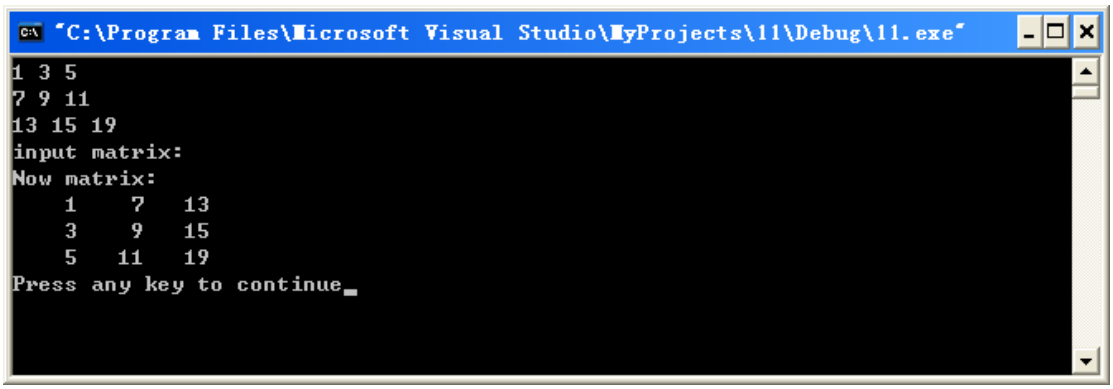
Take the address of the elements in row 0 and column 0 of the array as the function argument, and then realize the exchange of rows and columns during the function execution. After the function call, output the transposed two-dimensional array in the main function.

Code:

```
#include <stdio.h>
int main()
{
    void move(int *pointer);
    int a[3][3],*p,i;
    for(i=0;i<3;i++)
        scanf("%d%d%d",&a[i][0],&a[i][1],&a[i][2]);
    printf("input matrix:\n");
    p=&a[0][0];
    move(p);
    printf("Now matrix:\n");
    for(i=0;i<3;i++)
        printf("%5d%5d%5d\n",a[i][0],a[i][1],a[i][2]);
    return 0;
}

void move(int *pointer)
```

```
{
    int i,j,t;
    for(i=0;i<3;i++)
        for(j=i;j<3;j++)
        {
            t=*(pointer+3*i+j);
            *(pointer+3*i+j)=*(pointer+3*j+i);
            *(pointer+3*j+i)=t;
        }
}
```



Please consider:

① What does the pointer of a two-dimensional array, a line and an element mean? How should it be expressed?

Pointer of two-dimensional array: the first address of the first element of two-dimensional array

Pointer to a line: the first address of the first element of the line

Pointer to an element: the first address of the element

② How to represent the elements in row i and column j and their addresses?

Elements：a[i][j],*(a[i]+j),*(*(a+i)+j)

Address：a[i]+j,*(a+i)+j,&a[i][j]

Arrange the number of n in the reverse order of the input order, and use the function to realize

Use the array name as the function argument when calling the function.

The function argument is changed to a pointer to the first element of the array, and the formal parameter remains unchanged.

Analyze the similarities and differences between the above two.

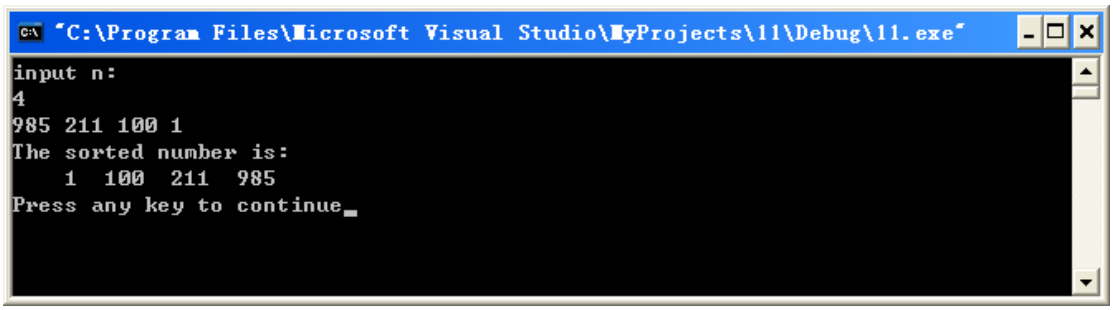
①

Code:

```
#include <stdio.h>
int main()
{
    void sort(int *p,int n);
    int i,n,num[20], *p;
    printf("input n:\n");
```

```
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&num[i]);
    }
    p=&num[0];
    sort(p,n);
    printf("The sorted number is:\n");
    for(i=0;i<n;i++)
        printf("%5d",num[i]);
    printf("\n");
    return 0;
}

void sort(int *p,int n)
{
    int i;
    int t,*p1,*p2;
    for(i=0;i<n/2;i++)
    {
        p1=p+i;
        p2=p+n-i-1;
        t=*p1;
        *p1=*p2;
        *p2=t;
    }
}
```

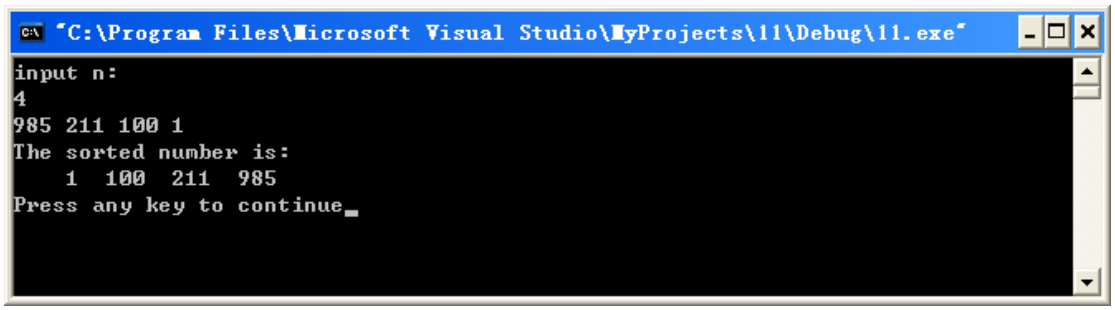


②

```
#include <stdio.h>
int main()
{
    void sort(int *p,int n);
    int i,n,num[20], *p;
    printf("input n:\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
```

```
    {
        scanf("%d",&num[i]);
    }
    sort(num,n);
    printf("The sorted number is:\n");
    for(i=0;i<n;i++)
        printf("%5d",num[i]);
    printf("\n");
    return 0;
}

void sort(int *p,int n)
{
    int i;
    int t,*p1,*p2;
    for(i=0;i<n/2;i++)
    {
        p1=p+i;
        p2=p+n-i-1;
        t=*p1;
        *p1=*p2;
        *p2=t;
    }
}
```



(4) Protocol function to find the length of a string. Enter a string in the main function and output its length

The following two cases shall be handled respectively in the program:

① The function parameter uses a pointer variable.

② The function parameter uses an array name.

Make analysis and comparison, and master its laws.

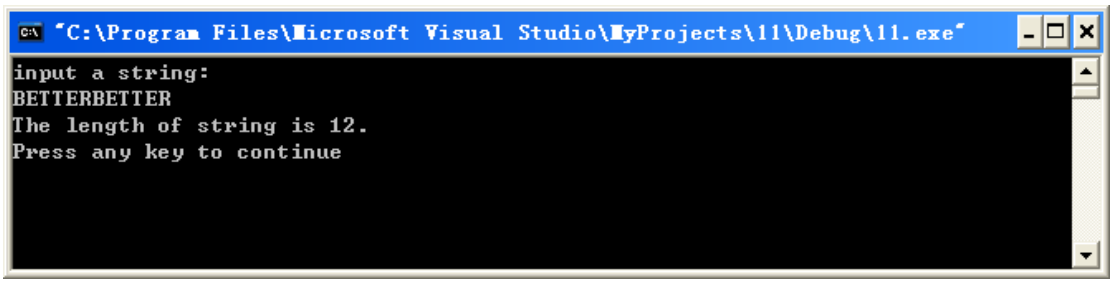
Code:

Pointer variable for function parameter：

```
#include <stdio.h>
int main()
{
```

```
    int length(char *n),len;
    char str[20];
    char *n;
    printf("input a string:\n");
    scanf("%s",str);
    n=&str[0];
    len=length(n);
    printf("The length of string is %d.\n",len);
    return 0;
}

int length(char *n)
{
    int m;
    m=0;
    while (*n!='\0')
    {
        n++;
        m++;
    }
    return (m);
}
```



②Array name for function parameter：

```
#include <stdio.h>
int main()
{
    int length(char *n),len;
    char str[20];
    printf("input a string:\n");
    scanf("%s",str);
    len=length(str);
    printf("The length of string is %d.\n",len);
    return 0;
}

int length(char *n)
{
```

```
    int m;
    m=0;
    while (*n!='\0')
    {
        n++;
        m++;
    }
    return (m);
}
```