采用数组在屏幕上显示杨辉三角形。

            1

            1     1

            1     2     1

            1     3     3      1

            1     4     6      4     1

            1     5     10     10    5     1

#include<stdio.h>

int main()

{

int i,j;//i为行数，j为每行的元素下标

int arr[7][6]={1};//定义二维数组并初始化

for(i=1;i<=6;i++)//i从1开始

{

arr[i][0]=1;

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

arr[i][j]=arr[i-1][j-1]+arr[i-1][j];

}

for(i=1;i<=6;i++)//i从1开始

{

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

printf("%-5d",arr[i][j]);//打印每一行

printf("\n");

}

return 0;

}

(2)有11个字母{q,w,e,r,t,y,u,M,o,p,e},利用数组进行排序并输出。要求用冒泡法和选择法两种方法实现排序，并分别记录两种排序方法的比较次数与交换次数，分析每一趟排序数组内容发生的变化。

/\*冒泡排序\*/

#include<stdio.h>

int main()

{

int i,j;

int sum1 = 0, sum\_1 = 0;//冒泡排序的比较和交换次数

char a[12] = { 'q','w','e','r','t','y','u','M','o','p','e'};

int flag = 0;//记录是否交换的标志变量

char k;

for (i = 1; i < 11; i++)

{

flag = 0;

for (j = 0; j < 11-i; j++)

{

flag = 0;

sum1 += 1;

if (a[j] > a[j + 1])

{

k = a[j];

a[j] = a[j + 1];

a[j + 1] = k;

flag = 1;

}

if (flag == 1)

sum\_1 += 1;

}

}

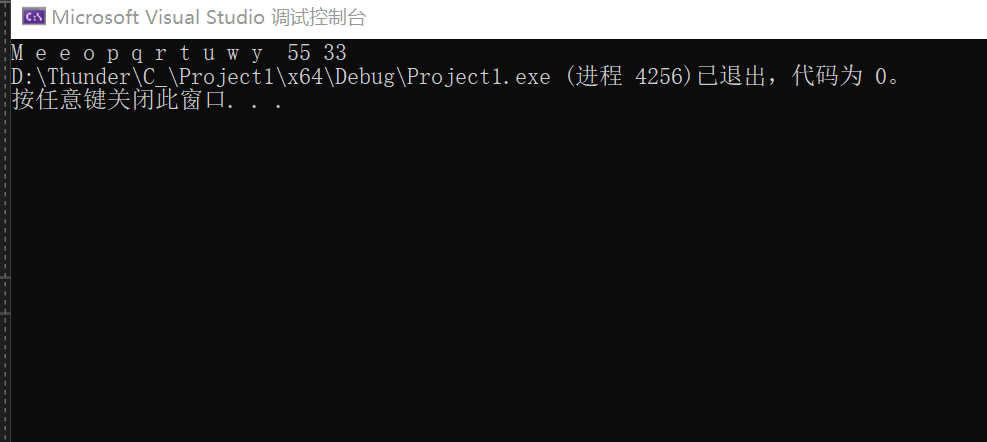
for (i = 0; i < 12; i++)

printf("%c ", a[i]);

printf("%d %d", sum1, sum\_1);

return 0;

}



/\*选择排序\*/

#include<stdio.h>

int main()

{

int i,j;

int sum1=0, sum\_1=0;//选择排序的比较和交换次数

char a[12] = {'q','w','e','r','t','y','u','M','o','p','e'};

int k;

char c;

for (i = 0; i < 11; i++)

{

k = i;//假设a[k]最小

for (int j = i + 1; j < 11; j++)

{

sum1++;

if (a[k] > a[j]) k = j;

}

if (k != i)

{

sum\_1++;

c = a[i];

a[i] = a[k];

a[k] = c;

}

}

for (i = 0; i < 11; i++)

printf("%c ",a[i]);

printf("%d %d\n",sum1,sum\_1);

return 0;

}

(3)有15个数按大小顺序存放在一个数组中，输入一个数，要求用折半查找法找出该数是数组中第几个元素的值。如果该数不在数组中，则输出“无此数”。

以15个数用赋初值的方法在程序中给出，要查找的数用scanf函数输入。

设算法的输入实例中有序的关键字序列为：(0,5,13,19,21,27,56,64,75,80,88,92,105,121,160)，要查找的关键字分别是21和85。

输出结果：

21   5

88   11

85   无此数

#include<stdio.h>

int main()

{

int b;

int left=0, right=14;

int mid=0;

int a[15] = { 0,5,13,19,21,27,56,64,75,80,88,92,105,121,160 };

printf("请输入要查找的数值：");

scanf("%d", &b);

while (left <= right)

{

mid = (left + right) / 2;

if (b > a[mid])

left = mid + 1;

else if (b < a[mid])

right = mid - 1;

else

{

printf("%d\n", mid+1);

break;

}

}

if (left > right)

printf("无此数\n");

return 0;

}

