

# 排序

# 排序

## 变量交换

4.3-1

```
#include<iostream>// 这段代码功能是交换 2 个数据
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int temp, a[10000]; // 预设保存 1 万个数的 a 数组; temp 是临时数据, 用完就丢
```

```
    int i;
```

```
    for (i=1; i<=2; i++)
```

```
        cin>>a[i];
```

```
    temp=a[1];
```

```
    a[1]=a[2];
```

```
    a[2]=temp;
```

```
    cout<<a[1]<<" "<<a[2];
```

```
    return 0;
```

```
}
```

```
3 5
5 3
```

# 选择排序

4.3-2

```
#include<iostream>// 选择排序
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n, i, j, temp, h[10000]; // 预设一个能保存1万个数据的a数组; temp是临时数据,  
    用完就丢
```

```
    cin>>n;
```

```
    for (i=1; i<=n; i++) cin>>h[i]; // 输入数据
```

```
    for (j=2; j<=n; j++) // 此段代码, 是找出最小数字, 放在下标1位置
```

```
    {
```

```
        if (h[j]<h[1])
```

```
        {
```

```
            temp=h[1];
```

```
            h[1]=h[j];
```

```
            h[j]=temp;
```

```
        }
```

```
        cout<<" 下标1与下标: "<<j<<" 比较: ";
```

```
        for (i=1; i<=n; i++) cout<<h[i]<<" ";
```

```
        cout<<endl;
```

```
    }
```

```
    cout<<endl<<" 最终排序: "<<endl;
```

```
    for (i=1; i<=n; i++) cout<<h[i]<<" ";
```

```
    return 0;
```

```
}
```

```
5  
12 7 26 5 3  
下标1与下标: 2 比较:  
7 12 26 5 3  
下标1与下标: 3 比较:  
7 12 26 5 3  
下标1与下标: 4 比较:  
5 12 26 7 3  
下标1与下标: 5 比较:  
3 12 26 7 5  
最终排序:  
3 12 26 7 5
```

#### 4.3-3

```
#include<iostream>// 选择排序
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n, i, j, temp, h[10000];
```

```
    cin>>n;
```

```
    for (i=1; i<=n; i++) cin>>h[i]; // 输入数据
```

```
    for (i=1; i<n; i++) // 从 1 到 n-1, 挨个拿出来比较大小 (最后一个数字不用排序?)
```

```
    {
```

```
        for (j=i+1; j<=n; j++) // 逐个与后面数字比较大小
```

```
        {
```

```
            if (h[j]<h[i])
```

```
                { temp=h[i]; h[i]=h[j]; h[j]=temp; }
```

```
        }
```

```
        cout<<endl<<"i="<<i<<": "<<" 从第 "<<i<<" 个数字到数字 "<<n<<" 找出最小  
数字放在下标 "<<i<<" 位置 :";
```

```
        cout<<h[1]<<" "<<h[2]<<" "<<h[3]<<" "<<h[4]<<" "<<h[5];
```

```
    }
```

```
    cout<<endl<<" 最终排序: "<<endl;
```

```
    for (i=1; i<=n; i++) cout<<h[i]<<" ";
```

```
    return 0;
```

```
}
```

5

12 7 26 5 3

i=1: 从第 1 个数字到数字 5 找出最小数字放在下标 1 位置 :3 12 26 7 5

i=2: 从第 2 个数字到数字 5 找出最小数字放在下标 2 位置 :3 5 26 12 7

i=3: 从第 3 个数字到数字 5 找出最小数字放在下标 3 位置 :3 5 7 26 12

i=4: 从第 4 个数字到数字 5 找出最小数字放在下标 4 位置 :3 5 7 12 26

3 5 7 12 26

#### 4.3-4

```
#include<iostream>// 选择排序
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n, i, j, temp, h[10000]; // 预设一个能保存1万个数据的a数组; temp是临时数据, 用完就丢
```

```
    cin>>n;
```

```
    for (i=1; i<=n; i++) cin>>h[i]; // 输入数据
```

```
    for (i=1; i<n; i++) // 从1到n-1, 挨个拿出来比较大小
```

```
    {
```

```
        for (j=i+1; j<=n; j++) // 逐个与后面数字比较大小
```

```
        {
```

```
            if (h[j]<h[i])
```

```
            {
```

```
                temp=h[i];
```

```
                h[i]=h[j];
```

```
                h[j]=temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    for (i=1; i<=n; i++) cout<<h[i]<<" ";
```

```
    return 0;
```

```
}
```

```
5
12 7 26 5 3
3 5 7 12 26
```

# 冒泡排序

4.3-5

#include<iostream>// 本段代码作用是找出最大的数字放在最后的位置

using namespace std;

int main()

{

int temp, a[10000]; // 预设一个能保存 1 万个数据的 a 数组; temp 是临时数据

int n, i, j;

cin>>n; // 输入数字个数

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

cin>>a[i]; // 挨个输入数字, 空格隔开

cout<<" 最初数据排列顺序: " <<endl; // 本句代码用于观察

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

{

cout<<a[i]<<" "; // 观察输入的数字顺序

}

////////////////////////////////////// 以下是冒泡排序代码

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

{

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

{

temp=a[j];

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

a[j+1]=temp;

}

cout<<endl;

cout<<" 第 "<<j<<" 次冒泡排序: " <<a[1]<<" " <<a[2]<<" " <<a[3]<<"

" <<a[4]<<" " <<a[5];

}

////////////////////////////////////// 以下是输出数组

cout<<endl<<" 排序结束: " <<endl; // 本句代码用于观察

for (i=1; i<=n; i++) cout<<a[i]<<" ";

return 0;

}

5

12 7 26 5 3

最初数据排列顺序:

12 7 26 5 3

第 1 次冒泡排序: 7 12 26 5 3

第 2 次冒泡排序: 7 12 26 5 3

第 3 次冒泡排序: 7 12 5 26 3

第 4 次冒泡排序: 7 12 5 3 26

排序结束:

7 12 5 3 26

#### 4.3-6

```
#include<iostream>// 冒泡排序
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int temp, a[10000];
```

```
    int n, i, j;
```

```
    cin>>n;// 输入数字个数
```

```
    for (i=1; i<=n; i++)
```

```
        cin>>a[i];// 挨个输入数字，空格隔开
```

```
////////////////////////////////////// 以下是冒泡排序代码
```

```
    for (i=n; i>1; i--)// 从最后一个数字开始，到第二个数字（第一个数字为什么不用排序？）
```

```
    {
```

```
        for (j=1; j<i; j++)// 从第一个数字开始，到本次循环倒数第二个数字（最后一个数字为什么不用排序？）
```

```
        {
```

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

```
            {
```

```
                temp=a[j];
```

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

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

```
            }
```

```
        }
```

```
        cout<<endl<<" 从第 "<<i<<" 个数字到数字 1 找出最大数字放在下标：  
"<<i<<" 位置 "<<endl;
```

```
        for (i=1; i<=n; i++)cout<<a[i]<<" ";
```

```
    }
```

```
////////////////////////////////////// 以下是输出数组
```

```
    cout<<endl<<" 排序结束: "<<endl;
```

```
    for (i=1; i<=n; i++)cout<<a[i]<<" ";
```

```
    return 0;
```

```
}
```

```
5  
12 7 26 5 3
```

```
从第 5 个数字到数字 1 找出最大数字放在下标： 5 位置
```

```
7 12 5 3 26
```

```
从第 4 个数字到数字 1 找出最大数字放在下标： 4 位置
```

```
7 5 3 12 26
```

```
从第 3 个数字到数字 1 找出最大数字放在下标： 3 位置
```

```
5 3 7 12 26
```

```
从第 2 个数字到数字 1 找出最大数字放在下标： 2 位置
```

```
3 5 7 12 26
```

```
排序结束：
```

```
3 5 7 12 26
```

#### 4.3-7

```
#include<iostream>// 冒泡排序
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int temp, a[10000]; // 预设一个能保存 1 万个数据的 a 数组; temp 是临时数据, 用完就丢
```

```
    int n, i, j;
```

```
    cin>>n; // 输入数字个数
```

```
    for (i=1; i<=n; i++)
```

```
        cin>>a[i]; // 挨个输入数字, 空格隔开
```

```
////////////////////////////////////// 以下是冒泡排序代码
```

```
    for (i=n; i>1; i--)
```

```
    {
```

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

```
        {
```

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

```
            {
```

```
                temp=a[j];
```

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

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

```
            }
```

```
        }
```

```
    }
```

```
////////////////////////////////////// 以下是输出数组
```

```
    cout<<endl<<" 排序结束 : "<<endl;
```

```
    for (i=1; i<=n; i++) cout<<a[i]<<" ";
```

```
    cout<<endl<<" 第 3 小的数字 : "<<a[3];
```

```
    cout<<endl<<" 第 2 大的数字 : "<<a[4];
```

```
    return 0;
```

```
}
```

```
5
12 7 26 5 3
排序结束 :
3 5 7 12 26
第 3 小的数字 :7
第 2 大的数字 :12
```



#### 4.3-8

```
#include<iostream>// 冒泡排序
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int temp, a[10000]; // 预设一个能保存 1 万个数据的 a 数组; temp 是临时数据, 用完就丢
```

```
    int n, i, j, min, max;
```

```
    cin>>n; // 输入数字个数
```

```
    for (i=1; i<=n; i++)
```

```
        cin>>a[i]; // 挨个输入数字, 空格隔开
```

```
    cin>>min; // 输入第几小的数字;
```

```
    cin>>max; // 输入第几大的数字
```

```
////////////////////////////////////// 以下是冒泡排序代码
```

```
    for (i=n; i>1; i--)
```

```
    {
```

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

```
        {
```

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

```
            {
```

```
                temp=a[j];
```

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

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

```
            }
```

```
        }
```

```
    }
```

```
////////////////////////////////////// 以下是输出数组
```

```
    cout<<endl<<" 排序结束 : "<<endl;
```

```
    for (i=1; i<=n; i++) cout<<a[i]<<" ";
```

```
    cout<<endl<<" 第 "<<min<<" 小的数字 "<<a[min];
```

```
    cout<<endl<<" 第 "<<max<<" 大的数字 "<<a[n-max+1];
```

```
    return 0;
```

```
}
```

```
5
12 7 26 5 3
3
2
```

```
排序结束 :
```

```
3 5 7 12 26
```

```
第 3 小的数字 7
```

```
第 2 大的数字 12
```