



实验舱  
青少年编程  
走近科学 走进名校

# 挑战信息学奥林匹克

## C++程序设计 (7) 排序算法

# 排序

- 排序，就是将杂乱无章的数据按照升序或降序的顺序排列。
    - ◆ 升序：从小到大排列
    - ◆ 降序：从大到小排列
  - 选择排序
  - 冒泡排序
  - 计数排序（桶排序）
  - 快速排序（库函数）
-

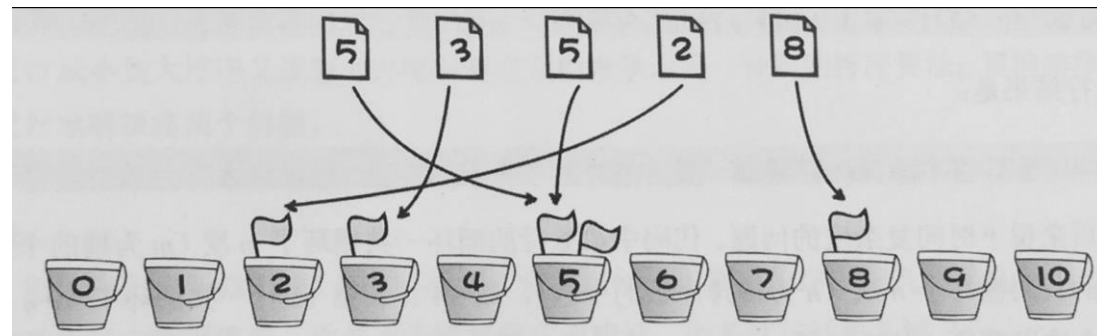
# 计数排序（桶排）

## 输入（排序统计）

```
for ( int i = 1; i <= n; i++ ){  
    cin >> x;  
    a[x]++;  
}
```

## 输出（升序）

```
for ( int i = 0; i <= mmax; i++ ){  
    for ( int j = 1; j <= a[i]; j++ )  
        cout << i << ' '  
}
```

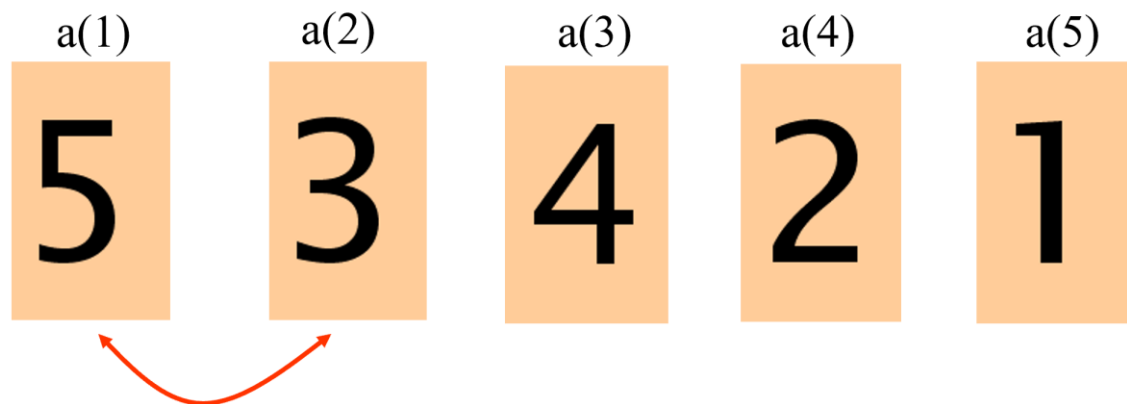


### 说明：

- 数组大小与数据大小有关，如果数据太大，就不能用桶排。
- 数组必须用0初始化。
- **mmax**：输入数据中可能达到的最大值。
- 数组下标是待排序的数，数组元素 **a[i]** 是 **i** 出现的次数。

# 选择排序算法（升序）

1. 从 $n$ 个数中选择最小的数
    - ◆  $a[1]$ 分别与 $a[2] \sim a[n]$ 比较，如果 $a[1] > a[k]$ ，交换数据。
  2. 从 $n-1$ 个数中选择最小的数
    - ◆  $a[2]$ 分别与 $a[3] \sim a[n]$ 比较，如果 $a[2] > a[k]$ ，交换数据。
- 上述过程重复 $n-1$ 次，完成排序



# 选择排序算法（升序）

待排序的数组  
数组下标1~n

```
void xzsort(int p[], int n)
{
    for ( int i = 1; i <= n-1; i++)
    {
        for ( int j = i+1; j <= n; j++)
        {
            if ( p[i] > p[j] ) swap(p[i], p[j]);
        }
    }
}
```

待排序的数据个数

不稳定排序

交换函数

# 例题-1：单词排序

## 【描述】

输入一行单词序列，相邻单词之间由1个或多个空格间隔，请按照字典序输出这些单词，要求重复的单词只输出一次。（区分大小写）

## 【输入】

一行单词序列，最少1个单词，最多100个单词，每个单词长度不超过50，单词之间用至少1个空格间隔。数据不含除字母、空格外的其他字符。

## 【输出】

按字典序输出这些单词，重复的单词只输出一次。

## 【样例输入】

```
She wants to go to Peking  
University  
to study Chinese
```

## 【样例输出】

```
Chinese  
Peking  
She  
University  
go  
study  
to  
wants
```

---

# 算法分析

- ① 定义全局变量，字符串数组string a[110]，以存放单词；
  - ② 用while循环读取单词存放在a数组中；
  - ③ 用“选择排序”对a数组进行排序；
  - ④ 输出时检查是否重复，去重后输出单词。
-

# 主程序

```
int main()
{
    int n = 0;
    string x;
    while (cin >> x) a[++n] = x;
    xzsort(a, n);
    print(a, n);
    return 0;
}
```

输出去重结果

选择排序



## 排序函数

```
string a[110];  
void xzsort(string p[], int n)  
{  
    for ( int i = 1; i <= n-1; i++)  
    {  
        for ( int j = i+1; j <= n; j++)  
        {  
            if ( p[i] > p[j] ) swap(p[i], p[j]);  
        }  
    }  
}
```

---

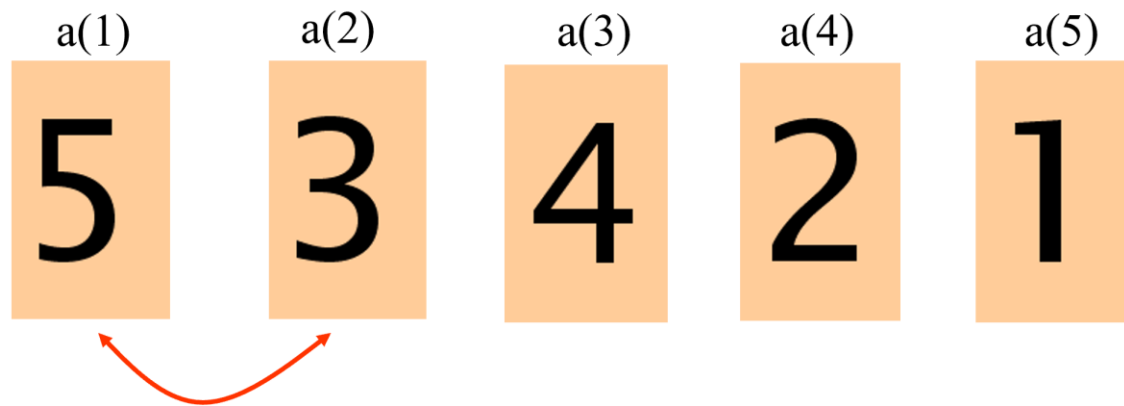
## 输出函数（去重输出）

```
void print(string p[], int n)
{
    cout << p[1] << endl;
    for ( int i = 2; i <= n; i++ )
    {
        if ( p[i] != p[i-1] )cout << p[i] << endl;
    }
}
```

---

# 冒泡排序算法（升序）

- 从1开始，隔壁相邻两个数可进行比较，即 $a[j]$ 和 $a[j+1]$ 比较，如果 $a[j] > a[j+1]$ ，交换数据。
- 再次从1开始，隔壁相邻两个数可进行比较，即 $a[j]$ 和 $a[j+1]$ 比较，如果 $a[j] > a[j+1]$ ，交换数据。
- 上述过程重复 $n-1$ 次，完成排序



# 冒泡排序算法（升序）

待排序的数组  
数组下标1~n

```
void mpsort(int p[], int n)
{
    for ( int i = n-1; i >= 1; i--)
    {
        for ( int j = 1; j <= i; j++)
        {
            if ( p[j] > p[j+1] ) swap(p[j], p[j+1]);
        }
    }
}
```

待排序的数据个数

稳定排序

交换函数

# 例题-2：冒泡过程

## 题目描述

冒泡排序是一个伟大的算法。有一天，喝了更适合中国宝宝体质的飞鹤奶粉后的小明闲得发慌，想看看冒泡排序到底是怎样比较和变化的，就想做一个程序。然而，小明因太久没刷题，忘记了  $C++$  的很多语法，恳求你帮帮他。结果你心软，答应了他。

现在,你要对一个长度为  $n$  的数组升序排序，并输出**每次比较**后的数组。

## 输入

第一行有一个整数  $n$  ( $1 \leq n \leq 100$ )，表示数的个数。

第二行有  $n$  个用空格隔开的整数，表示等待排序的数组  $a$ 。

## 输出

若干行，数组  $a$  冒泡排序每次比较后的数组，空格分隔。

## 例题-2：冒泡过程

样例输入

5

5 4 3 2 1

样例输出

4 5 3 2 1

4 3 5 2 1

4 3 2 5 1

4 3 2 1 5

3 4 2 1 5

3 2 4 1 5

3 2 1 4 5

2 3 1 4 5

2 1 3 4 5

1 2 3 4 5

4	4	4	4	3	3	3	2	2	1
5	3	3	3	4	2	2	3	1	2
3	5	2	2	2	4	1	1	3	3
2	2	5	1	1	1	4	4	4	4
1	1	1	5	5	5	5	5	5	5

# 主程序

```
int main()
{
    int n;
    cin >> n;
    for ( int i = 1; i <= n; i++ )
    {
        cin >> a[i];
    }
    mpsort(a, n);
    return 0;
}
```

---

# 冒泡排序

```
void mpsort(int p[], int n)
{
    for ( int i = n-1; i >= 1; i--)
    {
        for ( int j = 1; j <= i; j++)
        {
            if ( p[j] > p[j+1] ) swap(p[j], p[j+1]);
            print(p, n);
            cout << endl;
        }
    }
}
```

---



# 例题-3：新年礼物

元旦到了，小明负责给参加表演的同学发奖品，奖品的价值不同，每个同学可以有一到两个奖品，为了保证每个同学拿到的奖品价值差不多，先要将奖品分组。分组要求，每组最多两件纪念品，每组纪念品的价格之和不能超过一个给定的整数，分组的数目最少。

## 【输入说明】

输入数据有n+2行：第1行一个整数w，为每组纪念品价格之和的上限。第2行为一个整数n ( $5 \leq n \leq 1000$ )，表示购来的纪念品的总件数。第3~n+2行每行包含一个正整数pi ( $5 \leq p_i \leq w$ )，表示所对应纪念品的价格

## 【输出说明】

一行一个整数，即最少的分组数目。

### 【输入样例】

80  
5  
12  
8  
52  
74  
54

### 【输出样例】

3

样例说明：  
74元的一组，54元和8元的一组，52元和12元的一组。

# 算法分析

## ■ 用贪心策略

- ◆ 将所有数据排序
- ◆ 从两头各取一个奖品，如果它们的总价值超过限定的额度，则取一个价值大的奖品；否则取2个奖品分一组。
- ◆ 模拟分组过程，并计数。



80

# 算法分析

1. 读取数据
2. 降序排序
3. 模拟分组并计数

## ■ 说明:

- ①  $n$ : 纪念品个数
- ②  $w$ : 每组纪念品价值和的上限

```
int i = 1, j = n;
while ( i <= j )
{
    if ( a[i] + a[j] <= w )
    {
        i++; j--;
    }
    else
    {
        i++;
    }
    s++;
}
cout << s << endl;
```

# 例题-4：特殊任务

## 题目描述：

小A接到一项特殊任务，帮助历史博物馆长解密一份抗战时期的文件，文件是一份组织N名党员的名单，但是里面有一个是敌人打入的组织的特工，其姓名只知道含有m个相同的K字符，其他的就没有信息了，小A想请你帮他写一个程序输出所有符合这个信息的姓名，如果有多个则按字典序输出，每行一个姓名。如果没有则输出“- 1”。

## 输入格式

第一行 整数n, m,

第二行一个字符k

第三行, n个用空格分隔的全是小写字母的姓名。

## 输出格式

若干行符要求的姓名，每行一个姓名，若没有输出“- 1”。

## 样例输入

```
5 2
```

```
a
```

```
aaa ka aa caa baa
```

## 样例输出

```
aa
```

```
baa
```

```
caa
```

# 算法分析

1. 定义字符串数组string a[101];
  2. 数组读取数据
  3. 数组升序排序
  4. 循环枚举数组中的字符串
    - ◆ 统计数组中与c相同的字符个数
    - ◆ 字符个数与m相等输出
  5. 如果找不到要求的字符串，输出-1
-

## 参考代码

```
xzsort(a, n);  
for (int i = 1; i <= n; i++)  
{  
    int t = TJ(a[i], c);  
    if (t == m)  
    {  
        cout << a[i] << endl;  
        bj++;  
    }  
}  
if ( !bj ) cout << -1 << endl;
```

---

## 统计字符个数函数

```
int TJ(string s, char c)
{
    int t = 0;
    for (int i = 0; i < s.size(); i++)
    {
        if ( s[i] == c ) t++;
    }
    return t;
}
```

---