

10 - 一维数组的应用

C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 7 月 20 日

目录

1 复习回顾

2 数组的部分遍历

3 数组计数

4 数组表示状态

5 数组下标维护

6 总结

数组的声明

- 数组的声明
 - 元素类型 数组名 [数组大小];
 - `int a[8];` // 定义了可以存放 8 个整数的数组 a

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
<code>int a[8];</code>	71	80	62	91	99	82	43	53

数组的声明

- 数组的声明
 - 元素类型 数组名 [数组大小];
 - `int a[8];` // 定义了可以存放 8 个整数的数组 a
 - 数组中的基本单元叫作“元素”
 - 数组大小定义为可能存放元素数量的最大值（可适当再大一些）

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
<code>int a[8];</code>	71	80	62	91	99	82	43	53

数组的声明

- 数组的声明
 - 元素类型 数组名 [数组大小];
 - `int a[8];` // 定义了可以存放 8 个整数的数组 a
 - 数组中的基本单元叫作“元素”
 - 数组大小定义为可能存放元素数量的最大值（可适当再大一些）
 - 建议声明数组为全局变量

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
<code>int a[8];</code>	71	80	62	91	99	82	43	53

数组元素的访问

- 数组元素的访问
 - 通过 **数组名 [下标]** 访问存储在数组中某一位置的值
 - 下标是指在元素在数组中的位置，从 0 开始
 - 下标的范围： $0 \sim L - 1$ (L 为数组大小)

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
int a[8];	71	80	62	91	99	82	43	53

数组元素的访问

- 数组元素的访问
 - 通过 **数组名 [下标]** 访问存储在数组中某一位置的值
 - 下标是指在元素在数组中的位置，从 0 开始
 - 下标的范围： $0 \sim L - 1$ (L 为数组大小)
 - 例如：声明数组 `double b[5];`
 - 可以使用的数组下标：0, 1, 2, 3, 4
 - 对应的可以使用的数组元素：`b[0]`, `b[1]`, `b[2]`, `b[3]`, `b[4]`
 - 可以使用变量作为数组下标：`b[i]` ($0 \leq i \leq 4$)

	<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>	<code>a[4]</code>	<code>a[5]</code>	<code>a[6]</code>	<code>a[7]</code>
<code>int a[8];</code>	71	80	62	91	99	82	43	53

数组的遍历

- 遍历：对数组中所有元素逐个访问一遍的过程称为遍历
- 遍历输入数组
 - 存储在下标 $0 \sim n - 1$ 中
 - ```
for (int i = 0; i < n; i++) {
 cin >> a[i];
}
```



# 数组的遍历

- 遍历：对数组中所有元素逐个访问一遍的过程称为遍历
- 遍历输入数组
  - 存储在下标  $0 \sim n - 1$  中
    - ```
for (int i = 0; i < n; i++) {  
    cin >> a[i];  
}
```
 - 存储在下标 $1 \sim n$ 中
 - ```
for (int i = 1; i <= n; i++) {
 cin >> a[i];
}
```

# 数组的遍历

- 遍历：对数组中所有元素逐个访问一遍的过程称为遍历
- 遍历输出数组
  - 存储在下标  $0 \sim n - 1$  中
  - ```
for (int i = 0; i < n; i++) {  
    cout << a[i] << " ";  
}  
cout << endl;
```

数组的遍历

- 遍历：对数组中所有元素逐个访问一遍的过程称为遍历
- 遍历输出数组
 - 存储在下标 $0 \sim n - 1$ 中

```
for (int i = 0; i < n; i++) {  
    cout << a[i] << " ";  
}  
cout << endl;
```
 - 存储在下标 $1 \sim n$ 中

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

例 10.1：求数组元素的最大值及所有位置

编程题

- 第一行输入一个整数 n ($1 \leq n \leq 100$), 表示有 n 个整数, 第二行输入 n 个整数 x ($-10^9 \leq x \leq 10^9$), 表示数组中的每个元素, 求这 n 个数组元素中的最大值及最大值出现的所有位置。第一行输出最大值, 第二行以空格间隔输出所有位置。

- 样例输入

9

2 9 9 7 9 2 4 5 8

- 样例输出

9

2 3 5

例 10.1：求数组元素的最大值及所有位置

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int pos = 0;
12    for (int i = 1; i <= n; i++) {
13        if (pos == 0 || a[i] > a[pos]) pos = i;
14    }
15    cout << a[pos] << endl;
16    for (int i = 1; i <= n; i++) {
17        if (a[i] == a[pos]) cout << i << " ";
18    }
19    cout << endl;
20
21    return 0;
22 }
```

例 10.1：求数组元素的最大值及所有位置

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int pos = 0;
12    for (int i = 1; i <= n; i++) {
13        if (pos == 0 || a[i] > a[pos]) pos = i;
14    }
15    cout << a[pos] << endl;
16    for (int i = 1; i <= n; i++) {
17        if (a[i] == a[pos]) cout << i << " ";
18    }
19    cout << endl;
20
21    return 0;
22 }
```

目录

1 复习回顾

2 数组的部分遍历

3 数组计数

4 数组表示状态

5 数组下标维护

6 总结

例 10.2：求前 k 个数组元素的和

编程题

- 第一行输入一个整数 n ($1 \leq n \leq 100$)，表示有 n 个数组元素，第二行输入 n 个整数 x ($-10^6 \leq x \leq 10^6$)，表示数组中的每个元素。再输入一个整数 k ($1 \leq k \leq n$)，求前 k 个数组元素的和。

- 样例输入

6

1 4 2 8 5 7

4

- 样例输出

15

例 10.2：求前 k 个数组元素的和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int k; // 表示从 a[1] 累加到 a[k]
12    cin >> k;
13    int sum = 0;
14    for (int i = 1; i <= k; i++) sum += a[i];
15    cout << sum << endl;
16
17    return 0;
18 }
```

例 10.2: 求前 k 个数组元素的和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int k; // 表示从 a[1] 累加到 a[k]
12    cin >> k;
13    int sum = 0;
14    for (int i = 1; i <= k; i++) sum += a[i];
15    cout << sum << endl;
16
17    return 0;
18 }
```

例 10.3：求部分数组元素的和

编程题

- 第一行输入一个整数 n ($1 \leq n \leq 100$)，表示有 n 个数组元素，第二行输入 n 个整数 x ($-10^6 \leq x \leq 10^6$)，表示数组中的每个元素。再输入两个整数 l, r ($1 \leq l \leq r \leq n$)，求数组中第 l 至第 r 个元素之和。
- 样例输入
6
1 4 2 8 5 7
4 5
- 样例输出
13

例 10.3：求部分数组元素的和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int l, r; // 表示从 a[l] 累加至 a[r]
12    cin >> l >> r;
13    int sum = 0;
14    for (int i = l; i <= r; i++) sum += a[i];
15    cout << sum << endl;
16
17    return 0;
18 }
```

例 10.3：求部分数组元素的和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int l, r; // 表示从 a[l] 累加至 a[r]
12    cin >> l >> r;
13    int sum = 0;
14    for (int i = l; i <= r; i++) sum += a[i];
15    cout << sum << endl;
16
17    return 0;
18 }
```

目录

1 复习回顾

2 数组的部分遍历

3 数组计数

4 数组表示状态

5 数组下标维护

6 总结

例 10.4: 投票选班长 I

编程题

- 6 年 A 班在选班长，有 5 位候选人，编号 1 ~ 5，共 n 名同学将进行投票。

第一行输入整数 n ($1 \leq n \leq 100$)，如上所述，第二行输入 n 个整数 x ($1 \leq x \leq 5$)，表示每位同学的投票结果。

按候选人的编号顺序输出 5 行，每行输出候选人编号及票数，用空格间隔，表示得票结果。

- 样例输入

```
6
1 1 4 5 1 4
```

- 样例输出

```
1 3
2 0
3 0
4 2
5 1
```

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例说明

1	2	3	4	5
0	0	0	0	0

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例说明

1	2	3	4	5
1	0	0	0	0

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例说明

1	2	3	4	5
2	0	0	0	0

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例说明

1	2	3	4	5
2	0	0	1	0

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例说明

1	2	3	4	5
2	0	0	1	1

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例说明

1	2	3	4	5
3	0	0	1	1

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例说明

1	2	3	4	5
3	0	0	2	1

例 10.4：投票选班长 I

编程题

- 样例输入

6

1 1 4 5 1 4

- 样例输出

1 3

2 0

3 0

4 2

5 1

- 样例说明

1	2	3	4	5
3	0	0	2	1

例 10.4：投票选班长 I

```
1 #include <iostream>
2
3 using namespace std;
4
5 int cnt[10]; // cnt[x] 表示候选人 x 的票数
6
7 int main() {
8     int n;
9     cin >> n;
10    int x;
11    for (int i = 1; i <= n; i++) {
12        cin >> x;
13        cnt[x]++;
14    }
15    for (int i = 1; i <= 5; i++) {
16        cout << i << " " << cnt[i] << endl;
17    }
18
19    return 0;
20 }
```


例 10.5: 投票选班长 II

编程题

- 6 年 A 班在选班长, 有 m 位候选人, 编号 $1 \sim m$, 共 n 名同学将进行投票。

第一行输入两个整数 n ($1 \leq n \leq 100$) 和 m ($1 \leq m \leq 10$), 如上所述, 第二行输入 n 个整数 x ($1 \leq x \leq m$), 表示每位同学的投票结果。

按候选人的编号顺序输出 m 行, 每行输出候选人编号及票数, 用空格间隔, 表示得票结果。

- 样例输入

```
6 6
1 1 4 5 1 4
```

- 样例输出

```
1 3
2 0
3 0
4 2
5 1
6 0
```

例 10.5：投票选班长 II

```
1 #include <iostream>
2
3 using namespace std;
4
5 int cnt[15]; // cnt[x] 表示候选人 x 的票数
6
7 int main() {
8     int n, m;
9     cin >> n >> m;
10    int x;
11    for (int i = 1; i <= n; i++) {
12        cin >> x;
13        cnt[x]++;
14    }
15    for (int i = 1; i <= m; i++) {
16        cout << i << " " << cnt[i] << endl;
17    }
18
19    return 0;
20 }
```

例 10.6：分数统计

编程题

- 6 年 A 班刚刚考完期中考，编写一个程序，帮助老师统计每种分数的得分人数。

第一行输入两个整数 n ($1 \leq n \leq 100$) 和 m ($1 \leq m \leq 100$)，表示有 n 名同学参加考试，满分为 m 分。第二行依次输入 n 个同学的成绩 x ($0 \leq x \leq m$)。

对于所有出现了的成绩，按照 0 到 m 的次序，输出分数以及对应的学生人数，每条记录一行，以空格间隔。

- 样例输入

```
4 100
42 85 71 42
```

- 样例输出

```
42 2
71 1
85 1
```

例 10.6：分数统计

```
1 #include <iostream>
2
3 using namespace std;
4
5 int cnt[105];
6
7 int main() {
8     int n, m;
9     cin >> n >> m;
10    int x;
11    for (int i = 1; i <= n; i++) {
12        cin >> x;
13        cnt[x]++;
14    }
15    for (int i = 0; i <= m; i++) {
16        if (cnt[i]) cout << i << " " << cnt[i] << endl;
17    }
18
19    return 0;
20 }
```

例 10.7: 梦中的统计

编程题

- 输入两个整数 a 和 b ($1 \leq a \leq b \leq 10^6$), 统计 a 到 b 之间每个数中每个数码 (0 ~ 9) 的出现次数。
- 样例输入
129 137
- 样例输出
1 10 2 9 1 1 1 1 0 1
- 样例说明
在 129 到 137 之间, 数码 0 出现了 1 次, 数码 1 出现了 10 次, 以此类推

例 10.7: 梦中的统计

```
1 #include <iostream>
2
3 using namespace std;
4
5 int cnt[10];          // cnt[d] 记录数码 d 出现的次数
6
7 int main() {
8     int a, b;
9     cin >> a >> b;
10    for (int i = a; i <= b; i++) {
11        int x = i;      // 使用 i 的副本 x 进行拆分
12        do {
13            cnt[x%10]++; // 当前最低位的数码个数加一
14            x /= 10;
15        } while (x != 0);
16    }
17    for (int i = 0; i < 10; i++) cout << cnt[i] << " ";
18    cout << endl;
19
20    return 0;
21 }
```

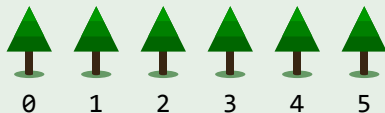
目录

- 1 复习回顾
- 2 数组的部分遍历
- 3 数组计数
- 4 数组表示状态**
- 5 数组下标维护
- 6 总结

例 10.8：校门外的树

编程题

- 某校大门外长度为 L ($1 \leq L \leq 10000$) 的马路上有一排树，每两棵相邻的树之间的间隔都是 1 米。在马路中，第一棵树在 0 的位置，最后一棵树在 L 的位置。例如 $L = 5$ 时，马路的情况如下图所示。



例 10.8：校门外的树

编程题

- 由于马路上有 m 个区域要建地铁，每个区域的起点和终点分别为 a, b ($0 \leq a \leq b \leq L$)，不同区域之间可能有重合的部分。现在要把这些区域中的树移走，对于每个区域，需要移走 $a \sim b$ 之间的树（包含端点 a 和 b ）。你的任务是计算将这些树都移走后，马路上还有多少棵树。

- 样例输入

10 3

2 3

5 7

6 8

- 样例输出

5

例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

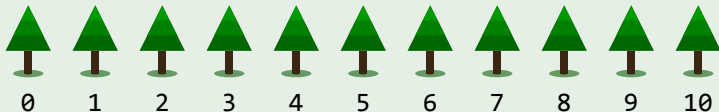
5 7

6 8

- 样例输出

5

- 样例说明



例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

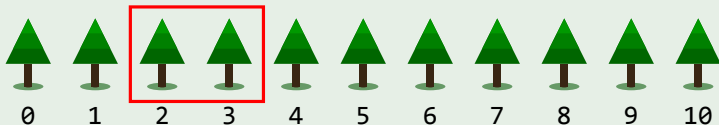
5 7

6 8

- 样例输出

5

- 样例说明



例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

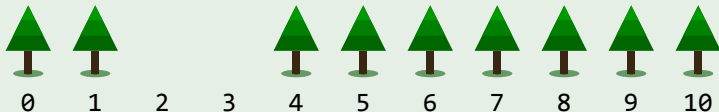
5 7

6 8

- 样例输出

5

- 样例说明



例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

5 7

6 8

- 样例输出

5

- 样例说明



0

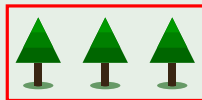
1

2

3



4



5

6

7



8



9



10

例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

5 7

6 8

- 样例输出

5

- 样例说明



例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

5 7

6 8

- 样例输出

5

- 样例说明



0

1

2

3



4

5



6

7



8



9



10

例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

5 7

6 8

- 样例输出

5

- 样例说明



0

1

2

3



4

5

6

7

8



9



10

例 10.8：校门外的树

如何表示以上过程？

例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

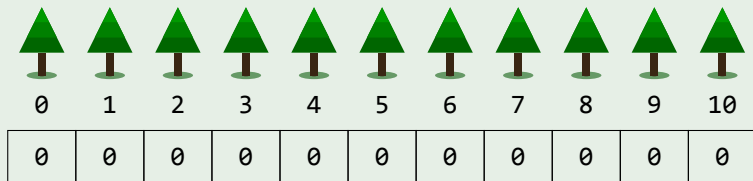
5 7

6 8

- 样例输出

5

- 使用数组记录状态



例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

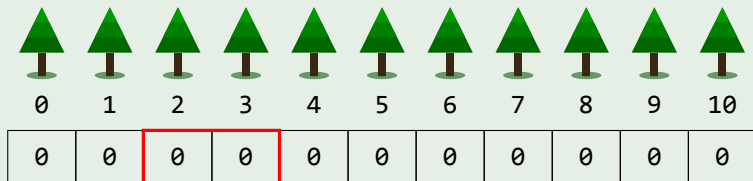
5 7

6 8

- 样例输出

5

- 使用数组记录状态



例 10.8: 校门外的树

编程题

- 样例输入

10 3

2 3

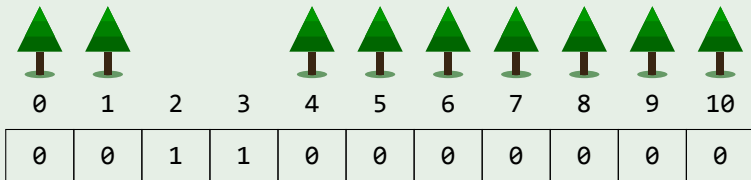
5 7

6 8

- 样例输出

5

- 使用数组记录状态



例 10.8: 校门外的树

编程题

- 样例输入

10 3

2 3

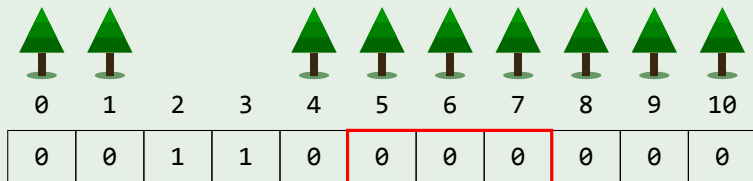
5 7

6 8

- 样例输出

5

- 使用数组记录状态



例 10.8：校门外的树

编程题

- 样例输入

10 3

2 3

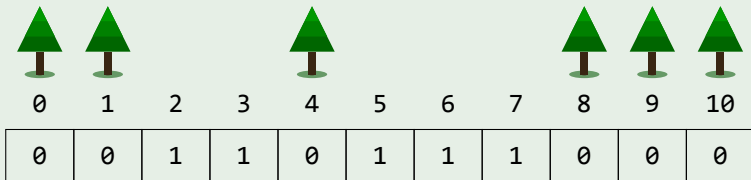
5 7

6 8

- 样例输出

5

- 使用数组记录状态



例 10.8: 校门外的树

编程题

- 样例输入

10 3

2 3

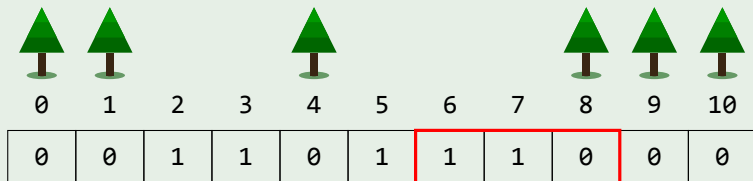
5 7

6 8

- 样例输出

5

- 使用数组记录状态



例 10.8: 校门外的树

编程题

- 样例输入

10 3

2 3

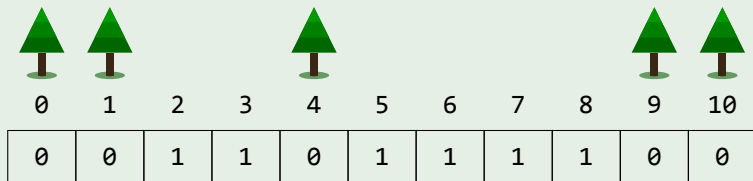
5 7

6 8

- 样例输出

5

- 使用数组记录状态



例 10.8: 校门外的树

```
1 #include <iostream>
2
3 using namespace std;
4
5 bool cut[10005];
6
7 int main() {
8     int l, m;
9     cin >> l >> m;
10    int a, b;
11    for (int i = 0; i < m; i++) {
12        cin >> a >> b; // 移除范围
13        for (int j = a; j <= b; j++) cut[j] = true; // 标记移除
14    }
15    int cnt = 0;
16    for (int i = 0; i <= l; i++) {
17        if (!cut[i]) cnt++; // 统计余下的树
18    }
19    cout << cnt << endl;
20
21    return 0;
22 }
```

目录

- 1 复习回顾
- 2 数组的部分遍历
- 3 数组计数
- 4 数组表示状态
- 5 数组下标维护**
- 6 总结

例 10.9：小鱼的数字游戏

编程题

- 输入若个数（不超过 100 个数），以 0 结尾，逆序输出所有输入的数。
- 样例输入
3 1 5 7 6 0
- 样例输出
6 7 5 1 3

例 10.9：小鱼的数字游戏

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n = 0, x;
9     while (cin >> x && x) {
10         n++;
11         a[n] = x;
12     }
13     for (int i = n; i >= 1; i--) {
14         cout << a[i] << " ";
15     }
16     cout << endl;
17
18     return 0;
19 }
```

例 10.9：小鱼的数字游戏

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n = 0, x;
9     while (cin >> x && x) {
10         n++;
11         a[n] = x;
12     }
13     for (int i = n; i >= 1; i--) {
14         cout << a[i] << " ";
15     }
16     cout << endl;
17
18     return 0;
19 }
```

目录

- 1 复习回顾
- 2 数组的部分遍历
- 3 数组计数
- 4 数组表示状态
- 5 数组下标维护
- 6 总结**

- 数组的应用
 - 数组的部分遍历
 - 数组计数
 - 数组表示状态
 - 数组下标维护

Thank you!