

全国青少年信息素养大赛-2026-初中组-算法-Cpp-星火-复赛-模拟题-01

一、单选题

(1) 边区兵工厂的弹药传送带是一个环形流水线，一头装填弹药、一头取走成品。我们用 `front` 指向当前可取弹药的队头位置，`rear` 指向队尾元素的下一个位置（即下一个可装填弹药的空闲位置），传送带最大容量为 `maxSize`。判断传送带已满、不能再装填的条件是（ ）。

- A. `rear == front`
- B. `(rear + 1) % maxSize == front`
- C. `(rear - 1 + maxSize) % maxSize == front`
- D. `(rear - 1) == front`

(2) 报务员截获多份敌军电台频率，同一频率可能被多次截获。为去重并按频率值有序排列，使用 `set<int>` 存储已确认的敌军频率。阅读下面代码，输出结果是（ ）。

```
1  #include <iostream>
2  #include <set>
3  using namespace std;
4
5  int main() {
6      set<int> freq;
7      freq.insert(45);
8      freq.insert(12);
9      freq.insert(45);
10     freq.insert(78);
11     freq.insert(12);
12     cout << freq.size() << " ";
13     cout << *freq.begin() << " ";
14     cout << *freq.rbegin();
15     return 0;
16 }
```

- A. 3 12 78
- B. 5 45 12

C. 3 45 78

D. 5 12 78

(3) 某作战部队的花名册采用双向链表管理，以便快速增删调动人员。现需将一名已调离的战士从链表中移除。下面代码实现该删除操作，横线处应填写（ ）。

```
1 void deleteNode(DoublyListNode*& head, int value) {  
2     DoublyListNode* current = head;  
3     while (current != nullptr && current->val != value)  
4         current = current->next;  
5     if (current != nullptr) {  
6         if (current->prev != nullptr) {  
7             _____ // 在此处填入代码  
8         } else {  
9             head = current->next;  
10        }  
11        if (current->next != nullptr)  
12            current->next->prev = current->prev;  
13        delete current;  
14    }  
15 }
```

A. `current->prev->next = current->next;`

B. `current->next = current->prev;`

C. `delete current->next;`

D. `current->prev = current->next;`

(4) 多路部队的战果分别统计后，需合并为一份完整的战报，同时要求相等的歼敌数保持原有上报次序。归并排序正适合这一需求。下面关于归并排序的描述，正确的是（ ）。

A. 归并排序是一个不稳定的排序算法

B. 归并排序的时间复杂度在最优、最差和平均情况下都是 $O(n \log n)$

C. 归并排序需要额外的 $O(1)$ 空间

D. 对于输入数组 $\{12, 11, 13, 5, 6, 7\}$ ，代码输出结果为 7 6 5 13 12 11

(5) 后勤处将各连队的补给物资按连队编号登记造册，同一连队多次登记的以最新数据覆盖旧数据。这种"编号→物资"的映射关系恰好用 `map` 实现。下面代码完成登记与查询，横线处应填入（ ）。

```
1  #include <iostream>
2  #include <map>
3  using namespace std;
4
5  int main() {
6      map<int, int> supply;           // 连队编号 → 物资数量
7      supply[3] = 150;
8      supply[1] = 200;
9      supply[3] = 180;               // 3号连队物资更新，覆盖旧值
10     supply[2] = 120;
11
12     cout << _____;           // 输出登记过的不同连队个数
13     supply.erase(1);               // 1号连队已完成补给，移除记录
14
15     for (auto it = supply.begin(); it != supply.end(); it++)
16         cout << it->first << ":" << it->second << " ";
17     return 0;
18 }
```

- A. `supply.size()`
- B. `supply.length()`
- C. `supply.count()`
- D. `supply.capacity()`

二、多选题

(6) 边区后勤处在设计物资数据库时，需要为不同种类的数据分配合适的存储类型——人数用整数、重量用长整数、标签用字符、状态用布尔值。关于 C++ 中常见数据类型及其大小的说法，正确的有（ ）。

- A. `int` 类型占 4 字节，`long long` 类型占 8 字节
- B. `char` 类型占 2 字节
- C. `double` 类型占 8 字节
- D. `bool` 类型占 1 字节

(7) 行军计划的编排经常用到不同的循环结构：已知行军天数的用固定次数的循环，条件未知的用判断式循环，至少执行一次任务的用先执行后判断的循环。关于 C++ 中三种循环结构的特征，以下说法正确

的有（ ）。

- A. `for` 循环适合循环次数已知的场景
- B. `while` 循环在条件判断前至少执行一次循环体
- C. `do-while` 循环在条件判断前至少执行一次循环体
- D. `for` 循环的初始化部分中声明的变量，其作用域仅限于该 `for` 语句

(8) 电台通信中每个字母和数字都有固定的编码值（ASCII 码）。报务员需要熟练进行字符与编码之间的转换，例如将数字字符转为实际数值、判断字母大小写等。关于字符与 ASCII 码，以下说法正确的有（ ）。

- A. 字符 `'0'` 的 ASCII 码是 48
- B. 大写字母的 ASCII 码比对应小写字母大 32
- C. 将数字字符转换为对应的整数值，可以用 `ch - '0'`
- D. 判断字符 `ch` 是否为小写字母可以用 `ch >= 'a' && ch <= 'z'`

(9) 各军分区上报的战功数据需按歼敌数排序，但同属一个分区的部队在排序后应保持内部原有先后顺序，这就对排序算法的"稳定性"提出了要求。关于排序算法的稳定性，以下说法正确的有（ ）。

- A. 冒泡排序是稳定的排序算法，因为相等元素不会发生交换
- B. 插入排序是稳定的排序算法
- C. 选择排序是不稳定的排序算法，因为在交换最小值时可能改变相等元素的相对顺序
- D. 快速排序是稳定的排序算法

(10) 前线阵地上的弹药箱堆放遵循"后放上的先取用"原则，这正是栈（Stack）的数据结构思想。在作战行动中，每一次推进和回撤的路线记录也如同栈的压入和弹出。关于栈，以下说法正确的有（ ）。

- A. 栈是一种后进先出（LIFO）的数据结构
- B. 栈的基本操作包括 `push`（入栈）和 `pop`（出栈）
- C. 函数调用过程中，局部变量和返回地址通常存储在系统栈中
- D. 在 C++ STL 中，`stack` 的 `top()` 操作不仅返回栈顶元素，还会将该元素移除

三、编程题

(1) Y3722 即时战功评功

描述

某次战役后，各参战部队的战功数据陆续上报至指挥部。指挥部按上报顺序逐一评定即时立功分数线：设立功比例为 $w\%$ ，已上报 p 份战功数据时，计划评功人数为 $\max(1, \lfloor p \times w\% \rfloor)$ 。如有战功数值相同，则所有并列者均获评功，实际评功人数可能超过计划数。

请你模拟这一实时评功过程：每收到一份新战功数据，立即输出当前的评功分数线（即当前排名前 $w\%$ 的最低战功值）。

输入描述

第一行有两个整数 n, w （ $1 \leq n \leq 100,000$ ， $1 \leq w \leq 99$ ），分别代表战功数据总数与立功比例。第二行有 n 个整数，依次代表逐一上报的战功值（ $0 \leq \text{战功值} \leq 600$ ）。

输出描述

输出一行，包含 n 个非负整数，依次代表每份战功数据上报后的即时评功分数线。相邻两个整数间用一个空格分隔。

输入样例 1

```
1 10 60
2 200 300 400 500 600 600 0 300 200 100
```

输出样例 1

```
1 200 300 400 400 400 500 400 400 300 300
```

(2) Y3723 情报暗号核对

描述

地下党接头时使用暗号字符串 A 和 B （长度不超过 50，只含小写字母）。为避免因口音或抄写误差导致误判，规定：若 A 与 B 完全相同，或 A 通过删除一个字符、插入一个字符或修改一个字符能够变成 B ，则视为暗号匹配，输出 "similar"；否则输出 "not similar"。共有 T 组暗号待核对（ $T \leq 100$ ）。

输入描述

第一行一个整数 T ，接下来 T 行每行两个字符串 A 和 B 。

输出描述

对于每组数据，输出 "similar" 或 "not similar"，每组占一行。

输入样例 1

```
1 5
2 apple applee
3 apple appe
4 apple bpple
5 applee bpple
6 apple apple
```

输出样例 1

```
1 similar
2 similar
3 similar
4 not similar
5 similar
```

(3) Y3724 军需预算规划

描述

八路军某师后勤部记录了接下来 N （ $1 \leq N \leq 100,000$ ）天的每日军需开销。

输入描述

第一行包含两个整数 N, M ，用单个空格隔开。

输出描述

输出一个整数，即最大预算周期开销的最小值。

输入样例 1

```
1 7 5
2 100
3 400
4 300
5 100
6 500
7 101
8 400
```

输出样例 1

```
1 500
```

(4) Y3725 密电频率筛查

描述

报务员截获了 n 份敌军电报，按截获时间顺序编号为 $1 \sim n$ ，每份电报有一个频率值。为研判敌军是否在同一频率上频繁更换加密方式，参谋需要找出频率值相同的电报中，截获序号间隔最近的那一对。如果有多对间隔相同，输出其中左端点在数组中位置最靠前（即左端点下标最小）的那对对应的频率值；若所有频率值均互不相同，输出 **No**。

输入描述

第一行一个整数 n （ $1 \leq n \leq 10^5$ ）。第二行 n 个整数，依次表示每份电报的频率值（绝对值不超过 10^9 ）。

输出描述

输出一个整数，即符合条件的最小频率值；若无相同频率值，输出 **No**。

输入样例 1

```
1 7
2 1 3 2 1 2 3 1
```

输出样例 1

```
1 2
```