

# 全国青少年信息素养大赛-2026-初中组-算法- Cpp-智传民韵-复赛-模拟题-01-答案与解析

|   |   |   |   |   |     |     |     |     |     |
|---|---|---|---|---|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 | 6   | 7   | 8   | 9   | 10  |
| B | A | A | B | A | ACD | ACD | ACD | ABC | ABC |

## 一、单选题

(1) 民族舞蹈道具仓库的环形传送带用于传递演出道具，一头放入道具、一头取出道具。我们用 `front` 指向当前可取道具的队头位置，`rear` 指向队尾元素的下一个位置（即下一个可放入道具的空闲位置），传送带最大容量为 `maxSize`。判断传送带已满、不能再放入的条件是（ ）。

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

【正确答案】 B

【题目解析】

-  正确答案： B
-  核心考点：循环队列的队满判断——牺牲一个位置区分队空与队满。
-  关键辨析： A 选项 `rear == front` 是队空条件，不是队满。循环队列规定当 `(rear + 1) % maxSize == front` 时，即队尾再向后走一步就追上队头，此时判为队满。
-  易错提醒： C 和 D 为迷惑选项，混淆了下标前进与后退方向。牢记公式：队空 `front == rear`，队满 `(rear + 1) % maxSize == front`。

(2) 舞蹈编导记录多支民族民间舞蹈的编号，同一编号可能被多次记录。为去重并按编号值有序排列，使用 `set<int>` 存储已确认的舞蹈编号。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> dance;
7      dance.insert(45);
8      dance.insert(12);
9      dance.insert(45);
10     dance.insert(78);
11     dance.insert(12);
12     cout << dance.size() << " ";
13     cout << *dance.begin() << " ";
14     cout << *dance.rbegin();
15     return 0;
16 }

```

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

【正确答案】 A

【题目解析】

-  正确答案： A
-  核心考点：考察 `set` 的两大核心特性——自动去重 + 自动按 key 排序。
-  执行追踪： `insert(45)` → `{45}`； `insert(12)` → `{12, 45}`； `insert(45)` → 重复，忽略，仍为 `{12, 45}`； `insert(78)` → `{12, 45, 78}`； `insert(12)` → 重复，忽略。最终 `size() = 3`，`*begin()` 返回最小元素 12，`*rbegin()` 返回最大元素 78。
-  易错提醒： `set` 自动去重——相同元素多次 `insert()` 只保留一个。`begin()` 指向最小元素，`rbegin()` 指向最大元素。不要去数 `insert()` 调用次数来判断 `size()`，重复插入不增加元素数量。

(3) 某舞团的舞者名录采用双向链表管理，以便快速增删参演人员。现需将一名已离团的舞者从链表中移除。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;`

【正确答案】 A

【题目解析】

-  **正确答案：** A
-  **核心考点：** 考察双向链表删除操作的指针调整——需要同时处理前驱和后继。
-  **执行追踪：** 删除双向链表中的节点需要：(1) 若被删节点不是头节点，将其前驱的 `next` 指向其后继： `current->prev->next = current->next` ；(2) 若被删节点不是尾节点，将其后继的 `prev` 指向其前驱： `current->next->prev = current->prev` 。代码中横线处处理的是第一项（头节点的 `prev` 为 `nullptr`）。
-  **易错提醒：** 双向链表删除涉及两个指针的调整（前驱的 `next` 和后继的 `prev`），缺一不可。代码先处理前驱（横线处），再处理后继（后面已给出），两者配合完成删除。

(4) 多个民族舞蹈分团的评分分别统计后，需合并为一份完整的汇演评分表，同时要求评分相等的舞团保持原有上报次序。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

【正确答案】 B

【题目解析】

-  正确答案： B
-  核心考点：考察归并排序的基本性质——稳定性、时间/空间复杂度及排序结果。
-  选项分析：
  - A  错误，归并排序是稳定的排序算法——合并左右子数组时，相等元素优先取左侧，保持原有相对顺序。
  - B  正确，归并排序采用稳定二分策略，任何时候复杂度都是  $O(n \log n)$ 。
  - C  错误，归并排序合并阶段需要临时数组暂存结果，额外空间为  $O(n)$ ，不是  $O(1)$ 。
  - D  错误，归并排序将数组按升序排列，输出应为 5 6 7 11 12 13。
-  易错提醒：归并排序是三大稳定排序之一（冒泡、插入、归并）。其  $O(n)$  额外空间是主要代价，但在链表排序中可降至  $O(1)$ 。

(5) 舞团管理处将各分团的服装道具按分团编号登记造册，同一分团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;           // 分团编号 → 道具数量
7      supply[3] = 150;
8      supply[1] = 200;
9      supply[3] = 180;               // 3号分团道具更新，覆盖旧值
10     supply[2] = 120;
11
12     cout << _____;           // 输出登记过的不同分团个数
13     supply.erase(1);               // 1号分团已完成整理，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()`

【正确答案】 A

【题目解析】

-  正确答案： A
-  核心考点：考察 `map` 的基本操作——`size()`、键唯一性、`erase()`、遍历输出。
-  执行追踪：虽然执行了 4 次 `[]` 赋值，但 `supply[3]` 出现了两次，第二次将值从 150 更新为 180（覆盖旧值），因此 `size()` 返回 3（三个不同的键：1, 2, 3）。`erase(1)` 移除键 1，遍历输出为 `2:120 3:180`（`map` 自动按键升序排列）。
-  易错提醒：`map` 的 `[]` 运算符若键不存在则插入新元素，若键已存在则覆盖旧值——不会增加元素数量。`count(key)` 返回某键的出现次数（对 `map` 只能为 0 或 1），不能用于求总元素个数。`map` 没有 `length()` 或 `capacity()` 方法。

## 二、多选题

(1) 舞团管理处设计舞蹈道具数据库时，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

【正确答案】 A,C,D

【题目解析】

-  正确答案： ACD
-  核心考点：考察 C++ 基本数据类型的字节大小记忆。
-  选项分析：
  - A  正确，`int` 占 4 字节（32 位），`long long` 占 8 字节（64 位）。
  - B  错误，`char` 占 1 字节（8 位），存储 ASCII 码值。

- C  正确，`double` 占 8 字节（64 位），双精度浮点数。
- D  正确，`bool` 占 1 字节，虽然理论上 1 位就够了，但 C++ 中最小可寻址单元是 1 字节。
-  **易错提醒：** `char` 是 1 字节不是 2 字节；不同平台下数据类型大小可能略有差异，但以上为通用标准。

(2) 舞蹈汇演的节目编排经常用到不同的循环结构：已知表演场次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

【正确答案】 A,C,D

【题目解析】

-  **正确答案：** ACD
-  **核心考点：** 考察三种循环结构的适用场景与关键区别。
-  **选项分析：**
  - A  正确，`for` 将初始化、条件、步进集中书写，最适合“已知循环次数”的场景。
  - B  错误，“至少执行一次”是 `do-while` 的特性，`while` 是先判断后执行的。
  - C  正确，`do-while` 先执行循环体再判断条件。
  - D  正确，`for (int i = 0; i < n; i++)` 中 `i` 的作用域仅限于 `for` 语句内，循环结束后 `i` 不可访问。
-  **易错提醒：** 选择循环结构的原则——已知次数用 `for`，未知次数用 `while`，至少一次用 `do-while`。

(3) 舞蹈编排中每个舞种和舞者都有固定的编码值（ASCII 码）。编导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'`

【正确答案】 A,C,D

【题目解析】

-  正确答案: ACD
-  核心考点: 考察字符与 ASCII 码的基本关系。
-  选项分析:
  - A  正确, `'0'` 的 ASCII 码为 48, `'A'` 为 65, `'a'` 为 97。
  - B  错误, 小写字母的 ASCII 码比对应大写字母大 32 (即 `'a' - 'A' = 32`), 选项说法正好反了。
  - C  正确, `'5' - '0' = 53 - 48 = 5`, 这是字符转数字的标准方法。
  - D  正确, 利用 ASCII 码的连续性判断字符范围。
  -  易错提醒: 务必牢记 `'a'` 比 `'A'` 大 32, 即小写字母 ASCII 码更大。

(4) 各舞蹈分团上报的参演评分需按数值排序, 但同属一个分团的节目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

【正确答案】 A,B,C

【题目解析】

-  正确答案: ABC
-  核心考点: 考察常见排序算法的稳定性判断。
-  选项分析:
  - A  正确, 冒泡排序在相等元素时不交换, 保证稳定性。
  - B  正确, 插入排序将元素插入到已排序序列的合适位置时, 遇到相等元素会放在其后, 不会跨越。

- C  正确，选择排序如 `[5_1, 5_2, 3]` → 选 3 与 `5_1` 交换得 `[3, 5_2, 5_1]`，两个 5 相对顺序改变。
- D  错误，快速排序在分区过程中可能跨越相等元素，是不稳定的。
-  **易错提醒**：稳定排序：冒泡、插入、归并。不稳定排序：选择、快排、堆排序。

(5) 舞蹈道具仓库中的演出服装遵循"后放上的先取用"原则，这正是栈（Stack）的数据结构思想。在舞蹈编排中，每一次道具的放入和取出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

【正确答案】 A,B,C

【题目解析】

-  **正确答案**： ABC
-  **核心考点**：全面考察栈的基本概念、操作和应用。
-  **选项分析**：
- A  正确，栈是 LIFO 结构。
- B  正确，`push` 和 `pop` 是栈的两个核心操作。
- C  正确，函数调用时参数、返回地址和局部变量压入系统调用栈，返回时弹出——这正是"栈"名称的由来。
- D  错误，STL 中 `top()` 只返回栈顶元素的值（不删除），`pop()` 才移除栈顶元素但不返回其值。两者需要配合使用。
-  **易错提醒**：C++ STL `stack` 的 `pop()` 返回类型为 `void`（不返回值），这是为了异常安全。必须先 `top()` 获取值，再 `pop()` 移除。

## 四、编程题

(1) Y3722 即时舞级评定

【题目描述】

民族舞蹈汇演后，各参演节目的评分数据陆续上报至组委会。组委会按上报顺序逐一评定即时优秀分数

线：设优秀比例为  $w\%$ ，已上报  $p$  份评分数据时，计划评优节目数为  $\max(1, \lfloor p \times w\% \rfloor)$ 。如有评分数值相同，则所有并列者均获评优，实际评优数可能超过计划数。  
请你模拟这一实时评优过程：每收到一份新评分数据，立即输出当前的优秀分数线（即当前排名前  $w\%$  的最低评分值）。

【输入描述】

第一行有两个整数  $n, w$ （ $1 \leq n \leq 100,000$ ， $1 \leq w \leq 99$ ），分别代表评分数据总数与评优比例。第二行有  $n$  个整数，依次代表逐一上报的评分值（ $0 \leq \text{评分值} \leq 600$ ）。

【输出描述】

输出一行，包含  $n$  个非负整数，依次代表每份评分数据上报后的即时优秀分数线。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 舞蹈暗号核对

【题目描述】

民间舞蹈传承人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 汇演预算规划**

**【题目描述】**

民族舞蹈展演组委会记录了接下来  $N$  （  $1 \leq N \leq 100,000$  ）天的每日活动开销。

为便于管理，须将连续的天数划分为  $M$  （  $1 \leq M \leq N$  ）个预算周期，每天恰好归属于一个周期。

组委会希望合理规划每个周期包含的天数，使开销最多的那个周期的总开销尽可能小。

**【输入描述】**

第一行包含两个整数  $N, M$  ，用单个空格隔开。

接下来  $N$  行，每行包含一个 1 到 10,000 之间的整数，按顺序给出接下来  $N$  天里每天的开销。

**【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 舞种编号筛查

##### 【题目描述】

舞蹈采风队记录了  $n$  支民间舞蹈，按采集时间顺序编号为  $1 \sim n$ ，每支舞蹈有一个舞种编号。为研判是否有地区在同一舞种上频繁变换变体，编导需要找出舞种编号相同的舞蹈中，采集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）的那对对应的舞种编号；若所有舞种编号均互不相同，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，依次表示每支舞蹈的舞种编号（绝对值不超过  $10^9$ ）。

##### 【输出描述】

输出一个整数，即符合条件的最小舞种编号；若无相同编号，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
```