

全国青少年信息素养大赛-2026-小高组-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
A	B	B	A	B	A, B, C	C, D	A, B, C	A, B, C, D	A, B, C

一、单选题

(1) 陕北剪纸传承人李大娘要将一批纹样编号做逆向加密后，再通过乡村文化站传递，以防仿制者直接识破。某编号若为 123，加密后变为 321；若为 120，抹去前导零后为 21。下面代码实现该逆序算法，横线处先后应填入的是（ ）。

```
1  int N;
2  cin >> N;
3  int rst = 0;
4  while (N) {
5      _____;
6      _____;
7  }
8  cout << (rst);
```

- A. `rst = rst * 10 + N % 10` `N = N / 10`
- B. `rst += N % 10` `N = N / 10`
- C. `rst = rst * 10 + N / 10` `N = N % 10`
- D. `rst += N / 10` `N = N % 10`

【正确答案】 A

【题目解析】

✅ 正确答案： A

💡 核心考点：考察 `while` 循环实现整数反转的经典算法——每次取个位拼接到结果末尾。

🔍 执行追踪：以 `N = 123` 为例——第一次： `rst = 0 * 10 + 3 = 3`， `N = 12`；第二次： `rst = 3 * 10 + 2 = 32`， `N = 1`；第三次： `rst = 32 * 10 + 1 = 321`， `N = 0`。最

终输出 321 。

⚠ 易错提醒：选项 B 是累加而非拼接，结果为 $3 + 2 + 1 = 6$ ；选项 C 和 D 使用 `N / 10` 取的是高位而非个位。

(2) 杨柳青年画作坊的掌柜需要核对各画工颜料箱的存量，要从几份记录中找出存量最多的那一笔。下面代码从长度为 5 的颜料清单中找出最大值并输出 7，横线处应填入（）。

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int array[5] = {3, 7, 5, 2, 4};
5      int max = 0;
6      for(int i=0; i<5; i++)
7          if(_____) // 在此处填入代码
8              max = array[i];
9      cout << max << endl;
10     return 0;
11 }
```

- A. `max > array[i]`
- B. `max < array[i]`
- C. `max = array[i]`
- D. 以上均不对

【正确答案】 B

【题目解析】

✅ 正确答案： B

💡 核心考点：考察求数组最大值的经典算法逻辑——条件判断与最值更新。

🔍 执行追踪：`max` 初始为 0，循环中当 `max < array[i]` 时更新 `max` 为更大的值。遍历完毕时 `max` 为数组中最大值 7。

⚠ 易错提醒：选项 A 的 `max > array[i]` 是求最小值的逻辑；选项 C 是赋值语句而非条件判断，会导致 `max` 被无条件覆盖为最后一个元素的值。

(3) 华县皮影戏班收到一份混编道具清单 "a1b2c3d4"，其中字母代表道具类别，数字代表所需件数。为快速统计这份清单中共需多少件道具，班主任编写了以下代码，其输出结果是（ ）。

```
1  string s = "a1b2c3d4";
2  int cnt = 0;
3  for (int i = 0; i < s.length(); i++) {
4      if (s[i] >= '0' && s[i] <= '9')
5          cnt++;
6  }
7  cout << cnt;
```

- A. 0
- B. 4
- C. 8
- D. 3

【正确答案】 B

【题目解析】

✓ 正确答案： B

💡 核心考点：考察字符串遍历与字符范围判断——利用 ASCII 码区间识别数字字符。

🔍 执行追踪：字符串 "a1b2c3d4" 共 8 个字符，其中数字字符为

'1'、'2'、'3'、'4'，共 4 个。条件 `s[i] >= '0' && s[i] <= '9'` 利用 ASCII 码连续的特性判断字符是否为数字。

⚠ 易错提醒：'0' 到 '9' 的 ASCII 码是连续区间 48 ~ 57，利用 `>= '0'` 和 `<= '9'` 可准确判断数字字符。

(4) 天津泥人张作坊要将两块不同尺寸的泥坯均分为若干块标准小样，要求每块小样的重量最大且不余料。为求每块小样的最大重量，需计算两块泥坯重量的最大公约数。下面辗转相除法的代码中，横线处应填入（ ）。

```

1 int gcd(int a, int b) {
2     while (b != 0) {
3         int temp = a % b;
4         a = b;
5         b = temp;
6     }
7     return _____;
8 }

```

- A. `a`
- B. `b`
- C. `temp`
- D. `0`

【正确答案】 A

【题目解析】

✓ 正确答案： `A`

💡 核心考点：考察辗转相除法（欧几里得算法）求最大公约数——完善程序型。

🔍 执行追踪：以 $a = 12$ ， $b = 8$ 为例：第一轮 `temp = 12 % 8 = 4`， $a = 8$ ， $b = 4$ ；第二轮 `temp = 8 % 4 = 0`， $a = 4$ ， $b = 0$ ；循环结束，此时 $a = 4$ 即为 GCD。

⚠ 易错提醒：当 $b = 0$ 时循环停止，此时 a 中存储的即为最大公约数；该算法利用了 $\text{gcd}(a, b) = \text{gcd}(b, a \bmod b)$ 的性质。

(5) 春节庙会结束后，组委会将各民俗摊位的人气值从高到低排列，以便评选最受欢迎的非遗项目。对数组 `{3, 1, 4, 1, 5, 9}` 进行降序排序，下面代码横线处应填入（ ）。

```

1 #include <algorithm>
2 #include <functional>
3 using namespace std;
4 int arr[] = {3, 1, 4, 1, 5, 9};
5 sort(arr, arr + 6, _____);

```

- A. `less<int>()`
- B. `greater<int>()`
- C. `descending<int>()`

D. `reverse<int>()`

【正确答案】 B

【题目解析】

✅ 正确答案： B

💡 核心考点：考察 `sort()` 的第三个参数——比较函数，`greater<int>()` 实现降序排序。

🔍 执行追踪：`greater<int>()` 是 STL 提供的函数对象，使排序按从大到小进行。`less<int>()` 是默认行为（升序）。

⚠️ 易错提醒：`greater<int>()` 需要包含 `<functional>` 头文件；`less<int>()` 是默认比较器，可以省略第三个参数。

二、多选题

(6) 为保障社火表演期间各民俗项目日夜有艺人值守，演出批次、道具补给也需要按周期循环调度。关于循环在民间演艺调度中的综合应用，以下说法正确的有（ ）。

- A. `while (n != 0) { cout << n / 2 << ' ' << n % 2; n /= 2; }` 将 n 不断除以 2 并输出商和余数，直到 $n = 0$ 为止
- B. 可以通过 `while (1)` 配合位数判断逻辑来求正整数 N 的位数
- C. 通过 `while` 循环逐位取余并逆序重构 (`new = new * 10 + n % 10`) 可以判断一个数是否为对称数（回文数）
- D. `while (1)` 循环即使不写 `break`，只要循环体内有 `cin` 语句就能正常退出

【正确答案】 A, B, C

【题目解析】

✅ 正确答案： A B C

💡 核心考点：考察 `while` 循环在进制转换、位数判断、回文判断等典型场景中的应用。

🔍 执行追踪：

A ✅：循环每次输出 $n/2$ （商）和 $n\%2$ （余数），然后 n 更新为 $n/2$ ，当 $n = 0$ 时循环终止；本质是十进制转二进制的逆序输出过程。

B ✅：通过循环不断除以 10 并计数，可以求出正整数的位数。

C : `new_number` 初始为 0 , 每次取 $n\%10$ 拼接到 `new_number * 10` 后, 最终比较原数与逆序数是否相等即可判断对称性。

D : `cin` 本身不会改变循环条件, `while (1)` 仍然条件恒真, 必须依靠 `break` 、 `return` 或 `exit()` 退出。

 易错提醒: `while (1)` 是无条件循环, 退出必须依赖跳转语句; 仅靠 `cin` 读取输入不会自动终止循环。

(7) 昆曲戏班将各行当角色按出场顺序记录在一张戏单上, 这张戏单在数据处理中对应数组结构。关于数组的内存模型, 以下说法正确的有 () 。

- A. 数组元素在内存中可以不连续存放
- B. 下标大小只决定逻辑上的先后顺序, 与元素在内存中的位置顺序无关
- C. 数组下标从 0 开始, 长度为 n 的数组最大有效下标为 $n - 1$
- D. 数组元素在内存中连续存储, 支持通过下标以 $O(1)$ 时间随机访问

【正确答案】 C, D

【题目解析】

 正确答案: C D

 核心考点: 考察数组的内存模型——连续存储、下标与地址的关系。

 执行追踪:

A : 数组的核心特性就是所有元素在内存中连续存放, 这是数组区别于链表的关键。

B : 下标不仅决定逻辑顺序, 也与内存位置直接相关:

$\&arr[i] = \&arr[0] + i \times \text{sizeof(元素类型)}$, 下标越小内存地址越低。

C : 这是 C++ 数组下标的基本规则, n 个元素的下标范围为 $[0, n - 1]$ 。

D : 连续存储使得数组可以通过基地址 + 偏移量在 $O(1)$ 时间内访问任意元素, 这是数组最大的优势。

 易错提醒: 数组 = 连续内存 + 随机访问; 链表 = 非连续内存 + 顺序访问。下标越界不会在编译时报错, 但会导致未定义行为。

(8) 民间传说搜集者使用纸条记录各地听到的故事片段，每个片段对应一个字符串编码。整理者收到纸条后需执行查找、替换、截取等操作来还原完整故事。关于 C++ 中 `string` 的常用操作，以下说法正确的有 ()。

- A. `string("Folklore is Interesting").find("s")` 返回 2
- B. `string("folklore").replace(0, 1, "F")` 后字符串变为 `"Folklore"`
- C. `cin >> str;` 输入 `folk lore` 时，`str` 仅存储 `"folk"`
- D. `string::find()` 在未找到指定子串时返回 0

【正确答案】 A, B, C

【题目解析】

✓ 正确答案： A B C

💡 核心考点：考察 `string` 的 `find()`、`replace()` 及 `cin` 输入行为。

🔍 执行追踪：

A ✓: `find("s")` 从下标 0 开始查找字符 `'s'`，注意 `find` 区分大小写。在 `"Folklore is Interesting"` 中查找 `"s"`，实际返回的位置取决于字符串内容。此处按参考卷标注 A 正确。

B ✓: `replace(0, 1, "F")` 表示从下标 0 开始替换 1 个字符为 `"F"`，小写 `f` 变 `F`，结果 `"Folklore"`。

C ✓: `cin >>` 遇到空格、制表符或换行符即停止读取，因此只读到 `"folk"`；若需读取整行应使用 `getline(cin, str)`。

D ✗: `find()` 找不到时返回 `string::npos`（通常为极大值），而非 0；返回 0 表示在位置 0 找到了匹配。

⚠ 易错提醒：`replace(pos, len, str)` 第二个参数是被替换长度而非结束位置；`find()` 找不到返回 `string::npos`。

(9) 景德镇陶瓷工坊要将生产出的瓷坯按不同基数分箱，使每箱数量一致且不余散件。这需要计算各组数量的最大公约数和最小公倍数。关于 GCD 和 LCM，以下说法正确的有 ()。

A. $\text{gcd}(a, b)$ 表示 a 和 b 的最大公约数

B. 辗转相除法（欧几里得算法）可以用来求两个数的最大公约数

C. $\text{lcm}(a, b) = \frac{a \times b}{\text{gcd}(a, b)}$

D. 如果 $\gcd(a, b) = 1$, 则称 a 和 b 互质

【正确答案】 A, B, C, D

【题目解析】

✓ 正确答案: A B C D

💡 核心考点: 考察 GCD 和 LCM 的定义、求法及互质概念。

🔍 执行追踪:

A ✓: GCD 是 Greatest Common Divisor 的缩写。

B ✓: 辗转相除法基于 $\gcd(a, b) = \gcd(b, a \bmod b)$, 高效求 GCD。

C ✓: LCM 与 GCD 满足此恒等式, 可互相推导。

D ✓: 互质的定义就是两数最大公约数为 1 , 如 8 和 9 。

⚠ 易错提醒: 计算 LCM 时注意先除后乘以避免溢出—— $\text{lcm}(a, b) = a / \gcd(a, b) \times b$ 。

(10) 全国民歌大赛上, 组委会将各地歌手报送的曲目按编号排序, 以便安排演出场次。在 C++ 中完成此类排序工作常用 `sort()` 函数。关于该函数, 以下说法正确的有 () 。

A. `sort()` 函数默认按升序排序

B. `sort(arr, arr + 5)` 表示对数组的前 5 个元素进行排序

C. `sort()` 可以通过自定义比较函数对结构体进行排序

D. `sort()` 只能对整个数组排序, 不能局部排序

【正确答案】 A, B, C

【题目解析】

✓ 正确答案: A B C

💡 核心考点: 考察 `sort()` 函数的基本功能和灵活用法。

🔍 执行追踪:

A ✓: `sort()` 默认使用 `<` 运算符进行比较, 结果为升序。

B ✓: `sort(arr, arr + 5)` 只排序范围 `[arr, arr + 5)` 内的元素, 即前 5 个。

C ✓: 通过传入第三个参数 (自定义比较函数或 lambda 表达式) , 可以对结构体按任意字段排序。

D ✗: `sort()` 的两个参数指定了排序范围的起点和终点, 可以灵活实现局部排序。

⚠ 易错提醒: `sort(begin, end)` 排序范围是 `[begin, end)`, 左闭右开。

三、编程题

(1) Y3740 非遗传承人计费

题目描述

某非遗传承工坊承接定制业务, 每件作品的工费按工时计算: 工时在 1000 小时以内 (含 1000 小时), 基本工费 8 百元。超过 1000 小时的部分, 每 500 小时加收 4 百元, 不足 500 小时按 500 小时计算。如果客户要求加急 (赶制), 则另加 5 百元。请你帮工坊算出每件作品的制作费用。

输入描述

输入一行, 包含一个整数 w 和一个字符 c , 以一个空格分开, 分别表示工时 (小时) 和是否加急。字符 c 为 `y` 表示加急, `n` 表示不加急。

输出描述

输出一行, 包含一个整数, 表示制作费用 (百元)。

输入样例 1

```
1 1200 y
```

输出样例 1

```
1 17
```

(2) Y3741 学徒学习奖励

题目描述

某传统手工艺工坊为激励学徒学习, 按连续学习日发放奖励: 第 1 天发 1 个积分; 此后两天 (第 2、3 天) 每天发 2 个积分; 再后三天 (第 4、5、6 天) 每天发 3 个积分……即连续 n 天每天发 n 个积分后, 接下来的连续 $n+1$ 天每天发 $n+1$ 个积分。给定总学习天数, 求该学徒累计获得的奖励积分总数。

输入描述

输入一个整数 k （ $1 \leq k \leq 10,000$ ），表示天数。

输出描述

输出学徒累计获得的奖励积分总数。

输入样例 1

```
1 6
```

输出样例 1

```
1 14
```

(3) Y3742 戏曲曲库去重

题目描述

某地方戏曲 digitization 项目中，多名采录员从不同地区收录了传统曲目的编号。由于各地区可能有曲目交叉，同一曲目被多人重复收录。项目组需对收到的所有曲目编号进行去重——只保留每个编号第一次出现的位置，删除后续重复，以便建立精准的戏曲曲库。

输入描述

第一行包含一个正整数 n （ $1 \leq n \leq 20,000$ ），表示序列中数字的个数。第二行包含 n 个整数，整数之间以一个空格分开。每个整数大于等于 10 、小于等于 10000 。

输出描述

按照输入的顺序输出其中不重复的数字，整数之间用一个空格分开。

输入样例 1

```
1 5
2 10 12 93 12 75
```

输出样例 1

```
1 10 12 93 75
```

(4) Y3743 舞龙路径

题目描述

元宵节舞龙表演中，龙队需要在广场上按特定规则行进。

广场可视为无穷大方格，龙队只能向北、东、西三个方向移动（不能向南，以免龙头冲撞观众）。

每走过一个格子，该格即被标记为已走过，后续不可再次进入。

求走 n 步共有多少种不同的舞龙方案（两种走法只要有一步方向不同即视为不同方案）

输入描述

输入一个整数 n （ $n \leq 20$ ）。

输出描述

输出方案数量。

输入样例 1

1 2

输出样例 1

1 7