CCF 编程能力等级认证
Grade Examination of Software Programming

C++ 三级

2026 年 03 月

1 单选题（每题 2 分，共 30 分）

第 1 题 如果字符变量 `_1` 的值是字符 `1`,那么 `(int) _1` 的值是? （ ）。

- A. 1
- B. -1
- C. 49
- D. +1 或者 -1

第 2 题 a,b是整型变量，各自有互不相同的初始值。下列程序实现了什么效果（ ）。

```
a=a^b;
b=a^b;
a=a^b;
```

- A. a,b 的值从始至终都没有改变。
- B. a,b 的值实现了互换。
- C. a,b 的值互换了以后，又还回去了，相当于没有变化。
- D. a,b 的值最后和原值不一样，没有任何意义。

第 3 题 关于下列正确的程序段，说法正确的是（ ）。

```
char str1[] = "Hello";
char str2[] ={'H','e','l','l','0'};
```

- A. 字符数组 `str1` 和 `str2` 完全相同。
- B.

```
cout<<str1<<endl;
cout<<str2<<endl;
```

这段程序将输出不同的结果。

- C. 字符数组 `str1` 和 `str2` 不相等。
- D. 这两个赋值方式完全相同。

第 4 题 关于以下程序段,说法正确的是（ ）。

```cpp
int x=10;
cout<<(x++) + (++x)<<endl;
```

- [ ] **A.** C++11 标准中，这是未定义行为，不同的环境有可能出现不同的结果
- [ ] **B.** 22
- [ ] **C.** 21
- [ ] **D.** 20

第 5 题 8 位二进制下，十进制数-15 的补码是（ ）。

- [ ] **A.** 11110000
- [ ] **B.** 10001111
- [ ] **C.** 10010000
- [ ] **D.** 11110001

第 6 题 三进制数 $2102_{(3)}$ 转换成十进制是：（ ）。

- [ ] **A.** 63
- [ ] **B.** 65
- [ ] **C.** 67
- [ ] **D.** 69

第 7 题 二进制数 10110101 是某数的 8 位补码，该数的十进制是（ ）。

- [ ] **A.** -73
- [ ] **B.** -75
- [ ] **C.** -77
- [ ] **D.** 75

第 8 题 已知 `unsigned char c = 0x0F;`（十六进制 `0F` = 二进制 `00001111`），执行 `c = c << 3;` 后，`c` 的十进制值是：（ ）。

- [ ] **A.** 64
- [ ] **B.** 72
- [ ] **C.** 80
- [ ] **D.** 120

第 9 题 补码的情况下，关于按位取反运算，用笔计算的情况下，以下说法错误的是：（ ）

- [ ] **A.** `~5` 的结果是 -6（`int` 类型，32 位）
- [ ] **B.** `~0` 的结果是 0（`int` 类型，32 位）
- [ ] **C.** `~(-3)` 的结果是 2（`int` 类型，32 位）
- [ ] **D.** `~8` 的结果是 -9（`int` 类型，32 位）

第 10 题 执行以下 C++ 代码后，`sub` 的值是（ ）。

```cpp
string str = "GESP2026";
string sub = str.substr(4, 2);
```

- [ ] **A.** 20
- [ ] **B.** 02
- [ ] **C.** 2026
- [ ] **D.** 026

第 11 题 执行以下代码后，输出结果是：（ ）。

```
int arr[] = {5, 10, 15, 20, 25, 30};
int count = sizeof(arr) / sizeof(arr[0]);
cout << count;
```

- [ ] **A.** 4
- [ ] **B.** 6
- [ ] **C.** 24
- [ ] **D.** 30

第 12 题 执行以下代码后，输出结果是：（ ）。

```
char s[10] = "abcde";
int a = sizeof(s) / sizeof(s[0]);
int b = strlen(s);
cout << a - b;
```

- [ ] **A.** 4
- [ ] **B.** 5
- [ ] **C.** 6
- [ ] **D.** 10

第 13 题 以下问题中，最不适合用枚举法解决的是：（ ）

- [ ] **A.** 找出 1~100 之间所有能被 7 整除的数
- [ ] **B.** 找出 100~200 之间的所有质数
- [ ] **C.** 计算 1+2+3+…+1000 的和
- [ ] **D.** 找出三位数中个位、十位、百位数字之和等于 10 的数

第 14 题 用枚举法解决“鸡兔同笼问题：头共 35 个，脚共 94 只，求鸡和兔的数量”，以下枚举逻辑最合理的是：（ ）

- [ ] **A.** 枚举鸡的数量 `x`（0~35），兔的数量 `y=35-x`，判断 `2*x + 4*y == 94`
- [ ] **B.** 枚举兔的数量 `y`（0~94），鸡的数量 `x=35-y`，判断 `2*x + 4*y == 94`
- [ ] **C.** 枚举所有整数 `x`（0-100）和 `y`（0-100），判断 `x+y==35 && 2*x+4*y==94`
- [ ] **D.** 枚举脚的总数 `sum`（0~94），判断 `sum == 94`

第 15 题 模拟“字符串加密”：规则为“每个字符 ASCII 码 + 3，若超过 `z` (122) 则从 `a` 重新开始”，以下代码中正确的条件判断是：（ ）

```
for (int i = 0; i < str.length(); i++) {
    // 需补充条件:
    ----------------------
    else {
        str[i] += 3;
    }
}
```

- [ ] **A.** `if (str[i]> 122) str[i] = str[i] + 3 - 26;`
- [ ] **B.** `if (str[i] > 122) str[i] = str[i] - 26;`
- [ ] **C.** `if (str[i] + 3 > 122) str[i] = str[i] - 23;`
- [ ] **D.** `if (str[i] + 3 > 'z') str[i] = 'a' + (str[i] + 3 - 'z') - 1;`

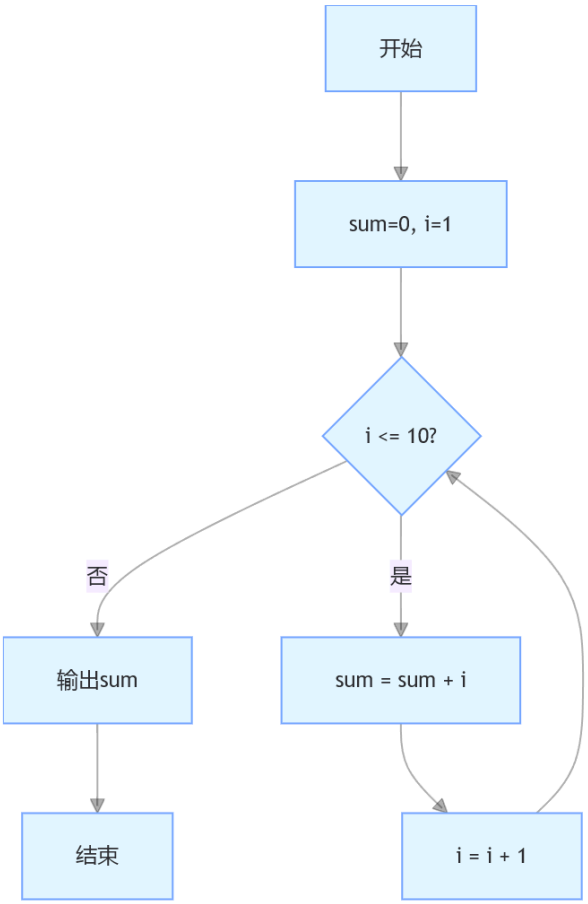
2　判断题（每题 2 分，共 20 分）

第 1 题 定义 `int arr[5] = {1,2,3};`，则 `arr[3]` 的值为 0，`arr[5]` 是合法下标。

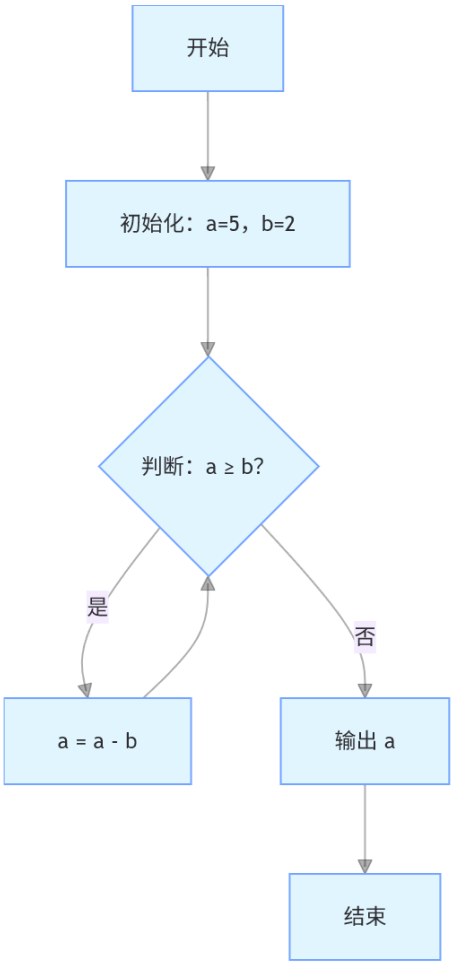
第 2 题 定义 `double arr[10];`，未手动初始化时，数组中所有元素的默认值为 0.0。

第 3 题 定义 `int arr[] = {1,2,3};`，则 `sizeof(arr)` 的结果为 12（`int` 占 4 字节）。

第 4 题 下面的流程图是用来求 1+2+3+…+10 的和。
请判断：这个流程图的逻辑正确还是错误?



第 5 题 下面流程图的功能是计算 5 对 2 取余数，输出结果为 1。



第 6 题 已知大写字符 `A` 的ASCII编码的十六进制表示为 `0x41` ，计算字符 `m` 的ASCII编码的八进制表示为 `155`（八进制）。

第 7 题 在 C++ 位运算中，各种不同的运算符有优先级的区分，使用括号能够解决优先级的问题。

第 8 题 由于在 0~255 范围内， `char` 类型和 `int` 类型可以互换，因此在这里 `x` 和 `y` 相等。

```
char x='1';
int y=1;
```

第 9 题 在C++语言中，表达式 `((0xf0 + 0x15) == 255)` 的值为 `true` 。

第 10 题 如果 `a` 为 `int` 类型的变量，且 `a` 的二进制最低位为 `0` ，则表达式 `((a & 3 & 1) == 0)` 的值为 `true` 。

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- **试题名称**：二进制回文串
- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.1.1 题目描述

对于一个正整数 n，我们将其转换为不含前导零的二进制表示,如果这个二进制序列从左向右读与从右向左读完全相同，则称该数为二进制回文数。例如，9 的二进制表示为 $(1001)_2$，是二进制回文数；12 的二进制表示为 $(1100)_2$，不是二进制回文数。

你的任务是：给定一个正整数 n，计算在 1 到 n 的范围内二进制回文数的数量。

3.1.2 输入格式

输入一行，包含一个正整数 n。

3.1.3 输出格式

输出一行，包含一个数，表示在 1 到 n 的范围内二进制回文数的数量。

3.1.4 样例

3.1.4.1 输入样例

```
15
```

3.1.4.2 输出样例

```
6
```

3.1.5 样例解释

样例 1 中，1 到 15 范围内 1、3、5、7、9、15 是二进制回文数。

3.1.6 数据范围

$1 \leq n \leq 10^5$。

3.2 编程题 2

- **试题名称**：凯撒密码
- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.2.1 题目描述

凯撒密码是一种替换加密技术，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例如，当偏移量是 3 的时候，所有的字母 A 将被替换成 D，B 被替换成 E，C 被替换成 F，以此类推，W 被替换成 Z，X 被替换成 A，Y 被替换成 B，Z 被替换成 C。这个加密方法是以罗马共和时期凯撒的名字命名的，据称当年凯撒曾用此方法与其将军们进行联系。

但是和所有的利用字母表进行替换的加密技术一样，凯撒密码非常容易被破解，而且在实际应用中也无法保证通信安全。

现在给你一个已破解的凯撒密码明文与密文，与一个有相同偏移量的未破解凯撒密码密文，请你帮忙破解它。

3.2.2 输入格式

输入共三行：

第一行包含一个字符串，表示已破解的凯撒密码明文；

第二行包含一个字符串，表示已破解的凯撒密码密文；

第三行包含一个字符串，表示待破解的凯撒密码密文。

3.2.3 输出格式

输出一行，包含一个字符串，表示待破解的凯撒密码对应的明文。

3.2.4 样例

3.2.4.1 输入样例

```
ABCDEFGVWXYZ
DEFGHIJYZABC
WKHTXLFNEURZQIRAMXPSVRYHUWKHODCBGRJ
```

3.2.4.2 输出样例

```
THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG
```

3.2.5 样例解释

样例 1 中，通过已破解的密码得出偏移量为 `'D' - 'A' = 3`，因此，对未破解部分进行逆向偏移：密文中的 `W` 对应明文中的 `T`（`'W' - 3 = 'T'`），密文中的 `K` 对应明文中的 `H`（`'K' - 3 = 'H'`），以此类推。

3.2.6 数据范围

保证密码长度均不超过 1000，所有字符串由大写字母组成。