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

C++ 一级

2026 年 03 月

1 单选题（每题 2 分，共 30 分）

第 1 题 2026年春节联欢晚会上一个武术表演节目《武 BOT》。节目中多个人形机器人会表演空翻，它们落地可能会有微微踉跄，但都会迅速调整姿态站稳，并适当移动来和前后左右的其他机器人保持原来队列。如果将机器人视作一个计算机系统，那么在该计算机系统中下面哪一项不能作为输入设备（ ）。

- A. 检测重心的重力传感器
- B. 预装的AI算法程序
- C. 接收动作指令的遥控器
- D. 拍摄其他机器人的摄像头

第 2 题 小明学习编程有一段时间了，他想在图形环境下把当前目录（或文件夹）下的文本文件 `20260314.txt` 的名字改一下。他用鼠标左键点击选中该文件后，立即完成下面哪个操作后将处于输入新文件名的状态（ ）：

- A. 单击右键并选择弹出菜单中的“重命名”
- B. 双击左键
- C. 按功能键 F1
- D. 按回车键

第 3 题 下面C++代码可以执行，有关说法正确的是（ ）。

```
double PI = 3.1415926;
cout << (PI);
```

- A. 为了方便初学者，`cout << (PI)` 和 `cout << (pi)` 效果相同，即变量的大小写不敏感
- B. `cout << (PI)` 修改为 `cout << (Pi)` 能正常执行
- C. 不能用 `PI` 做变量名，因为要保存圆周率这个常量
- D. 将程序中全部 `PI` 都改写为 `Pai`，将能正常执行，不会报错

第 4 题 C++表达式 `3 * 3 % 2` 的值为（ ）。

- A. 81
- B. 27
- C. 4
- D. 1

第 5 题 整型变量 `a` 、 `b` 的初值都是 4，则下面的C++代码执行后的输出是（　　）。

```
a, b = 3, 4;
cout << (a + 2) << (b - 2) << endl;
cout << a << b << endl;
```

- [ ] **A.**

```
61
43
```

- [ ] **B.**

```
52
34
```

- [ ] **C.**

```
62
44
```

- [ ] **D.**

```
62
32
```

第 6 题 下面C++代码的相关说法，正确的是（　　）。

```
int N = 0;
cin >> N;
cout << (N);
```

- [ ] **A.** 执行时如输入 `10` ，则将输出 `10`
- [ ] **B.** 执行时如输入 `3.14` ，将报错
- [ ] **C.** 执行时如输入 `ABC` ，将报错
- [ ] **D.** 执行时如输入 `-10` ，将报错

第 7 题 下面C++代码执行时，其说法正确的是（　　）。

```
int M = 0, N = 0;
cin >> M;
cin >> N;

if (N > M)
    cout << (N - M);
else
    cout << (M - N);
```

- [ ] **A.** 如果输入一个正数和一个负数，其输出结果肯定是大于0
- [ ] **B.** 不管是负整数、正整数亦或0，其结果肯定是大于等于0
- [ ] **C.** 如果 `N` 和 `M` 是相等的整数，将不会有输出
- [ ] **D.** 如果 `N` 和 `M` 输入带有小数点的数，将按整数部分计算

第 8 题 下面C++代码执行后的输出是（　　）。

```
int i, tnt = 1;
for (i = 0; i < 5; i++)
    tnt *= i;
printf("%2d%2d\n", tnt, i);
```

- [ ] **A.**

```
24 5
```

- [ ] **B.**

```
10 5
```

- [ ] **C.**

```
 0 4
```

- [ ] **D.**

```
 0 5
```

第 9 题 执行下面C++代码段求序列 `-1+2+3-4+5+6-7+8+9-10+11+12.....` 之值。例如输入 `4`，则计算序列前4项的值，规律如序列所示，输出为 `0`。下面说法中正确的是（　　）。

```
int N, tnt, i;
cout << "请输入正整数: ";
cin >> N;

tnt = 0;
for (i = 1; i < N + 1; i++)  // L1
    if (i % 3 == 1)          // L2
        tnt += -i;
    else
        tnt += i;
cout << tnt;
```

- [ ] **A.** L1行中 `i < N + 1` 应该修为 `i < N` 才会符合预期
- [ ] **B.** L2行中 `i % 3 == 1` 应修改为 `i % 3 == 0` 才会符合预期
- [ ] **C.** L2行中 `i % 3 == 1` 修改为 `i % 3` 与当前程序效果相同
- [ ] **D.** 当前代码能实现题目所描述计算目标

第 10 题 下面C++代码的相关说法，正确的是（　　）。

```
int i;
for (i = 1; i < 10; i++){
    if (i % 2 == 0){
        continue;       // L1
    }
    else
        cout << i << "#";
}
cout << i << "END";
```

- [ ] **A.** 上述代码执行后，其输出是 `1#3#5#7#9#9END`
- [ ] **B.** 删除 `else` 后的执行效果与当前代码相同
- [ ] **C.** 删除 `else` 且将 `cout << i << "#"` 移入L1行下面，则执行效果与当前代码相同
- [ ] **D.** 在 `cout << i << "END"` 前增加判断 `if(i > 10)`，其执行效果与当前代码相同

第 11 题 一个正整数的每位都是个位数，称为数位，最高位非0。下面的C++代码用于求正整数的所有数位之和，简称数位和。如 `123` 的各数位分别是1、2、3，则其数位和为 `1+2+3`，结果为6。为实现该目标，横线处应该填写的代码是（　　）。

```
int N;
cin >> N;
int tnt = 0;
while (N != 0){
    ____________;
    ____________;
}
cout << "N的数位和为: " << tnt;
```

- [ ] **A.**

```
tnt += N / 10
N /= 10
```

- [ ] **B.**

```
tnt += N % 10
N /= 10
```

- [ ] **C.**

```
tnt += N / 10
N %= 10
```

- [ ] **D.**

```
tnt = tnt + N % 10
N %= 10
```

第 12 题 小明想要快速知道任给一个正整数中有多少个奇数位(数位值是奇数)，下面的C++代码是其实现，横线处应该填入的代码是（　　）。

```
int N;
cin >> N;

int odd_count = 0;  // 记录奇数的个数
int old_number = N; // 保存原数

while (N != 0){
    if (__________________)
        odd_count += 1;

    N = (N - N % 10) / 10;
}
cout << old_number << "中共有 " << odd_count << " 个奇数";
```

- [ ] **A.** `N % 10 % 2 == 0`
- [ ] **B.** `N % 10 % 2 == 1`
- [ ] **C.** `N / 10 / 2 == 1`
- [ ] **D.** `N / 2 / 10 == 0`

第 13 题 小明和弟弟在玩一个拼数字游戏，游戏规则是：二人各写一个两位正整数 `M` 和 `N`；然后将较大的数字放在较小的前面，拼成一个4位数；将这个4位数除以3的余数添加到4位数的后面，得到一个5位数；最后判断这个5位数能否被14整除。下面的C++代码用于判断 `M` 和 `N` 是否符合全部要求，如果符合则输出 `Y` 否则输出 `N`。例如输入 `85` 和 `79`，可以拼出 `85792`,恰好是14的倍数，则输出 `Y`。关于下面代码描述正确的是（　　）。

```cpp
int M, N, Q;
cin >> M >> N;

if(M > N)    // L1
    Q = M*100+N;
else
    Q = N*100+M;

if((Q*10+Q%3) % 14 == 0)    // L2
    cout << "Y";
else
    cout << "N";
```

- [ ] **A.** 代码段不能完成正确判断
- [ ] **B.** L1行代码中条件应该改为 `M <= N`
- [ ] **C.** L2行代码应该改为 `if ((Q*10+Q/3) % 14 == 0)`
- [ ] **D.** 代码段可以不使用变量 `Q`，而是在区分 `M` 和 `N` 大小后分别直接用它们来判断

第 14 题 执行下面C++代码可以判断一个6位正整数 `N` 的高3位和低3位的差是否是314的倍数。例如 `628314` 就符合要求。横线处应该填入（　　）。

```cpp
cin >> N;
if (____________________________)
    cout << N << "符合条件" << endl;
```

- [ ] **A.** `((N % 1000) - (N / 1000)) / 314 == 0`
- [ ] **B.** `((N / 1000) - (N % 1000)) % 314 == 0`
- [ ] **C.** `((N % 1000) - (N / 1000)) / 314`
- [ ] **D.** `((N / 1000) - (N % 1000)) % 314`

第 15 题 `N` 是一个正整数。如果 `N` 的所有奇数位的数位和等于所有偶数位的数位和，则称它是一个“双螺旋数”。例如 `12375` 的所有奇数位的数位和是9，同时它的所有偶数位的数位和也是9，则 `12375` 就是一个双螺旋数。下面的C++代码用于判断输入的 `N` 是否为双螺旋数。空白处应该填入的代码是（　　）。

```cpp
int i, N, N1=0, N2=0, N0;
cin >> N;
N0 = N;
while (N){
    ------------------------
    ------------------------
}
if(N1 == N2)
    cout << N0 << "是双螺旋数" << endl;
```

- [ ] **A.**

```cpp
N1 += N%10, N /= 10;
N2 += N%10, N /= 10;
```

- [ ] **B.**

```cpp
N1 += N/10, N /= 10;
N2 += N/10, N /= 10;
```

☐ **C.**

```
N1 += N%10, N %= 10;
N2 += N%10, N %= 10;
```

☐ **D.**

```
N1 += N/10, N %= 10;
N2 += N/10, N %= 10;
```

2　判断题（每题 2 分，共 20 分）

第 1 题 小明的妈妈最近刚刚给他买了一块电话手表，除了可以看时间，小明也可以用它和妈妈打电话、收发信息，那么可以推测这块手表中装有一款特定操作系统。

第 2 题 C++表达式 `4 % 2` 和 `2 * 2 % 2` 的结果相同。

第 3 题 下面C++代码段成功执行后将输出 `0` 。

```
for (i = 1; i < 10; i++)
    if (i % 3 == 0)
        break;
cout << i;
```

第 4 题 下面能够正常执行的C++代码段用于求1到N之和，N为正整数。因为 `i < N + 1` ，所以是1到N且包含N之和。

```
tnt = 0;
cout << "请输入正整数: ";
cin >> N;
total = 0;
for (i =1; i < N + 1; i++) ;
    total += i;
cout << total;
```

第 5 题 执行下面的C++代码段，其语句 `cout << (N)` 将被执行0次或无数次（即死循环）。

```
cin >> N;
while (N)
    cout << (N);
```

第 6 题 下面的C++代码段的变量都是整型，它能用于判断输入的正整数是否为对称数。所谓对称数是指从左到右和从右到左读该数，其值相同。例如， `121` 和 `414` 都是对称数，而 `123` 不是对称数。（ ）

```
cout << "请输入正整数:";
cin >> n;

old_number = n;
new_number = 0;
while (n != 0){
    new_number = new_number * 10 + n % 10;
    n /= 10;
}

if (old_number == new_number)
    cout << "对称数";
else
    cout << "非对称数";
```

第 7 题 执行下面的C++代码段，如果变量都为整型变量，输入为大于0的整数，则输出数值一定为 `-N` 的值。

```
cin >> N;
total = 0;
for (i = -N; i < N; i +=2)
    total += i;
cout << total;
```

第 8 题 执行C++语句 `printf("%d\n", 3.14)` 将报错。

第 9 题 执行下面的C++代码后将输出 `2500` 。

```
int cnt = 0;
for (int i = 1; i < 100; i++)
    cnt += i++;
cout << cnt;
```

第 10 题 小明在测试C++的 `printf` 的功能时执行了 `printf("%-5d\n", 314)`，则代码输出的结果是 `-5314` 。

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- **试题名称**：交朋友
- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.1.1 题目描述

Alice 班上共有 4 个小朋友，身高分别为 H_1, H_2, H_3, H_4，其中 Alice 的身高为 H_1。

Alice 想要和身高最接近她的人交朋友，如果有多个人符合条件，则 Alice 想和其中较矮的那一人做朋友，你能告诉她这个人的身高是多少吗?

3.1.2 输入格式

输入共 4 行，第 i 行包含一个整数 H_i，表示班上小朋友的身高。

3.1.3 输出格式

输出 1 行，包含一个整数 h，表示 Alice 想交的朋友的身高。

3.1.4 样例

3.1.4.1 输入样例

```
150
165
135
133
```

3.1.4.2 输出样例

```
135
```

3.1.5 样例解释

样例 1 中，Alice 身高为 150，第 2、3 个小朋友与 Alice 身高差距为 15，同样最接近，Alice 选较矮的一个即第 3 个身高为 135 的小朋友交朋友。

3.1.6 数据范围

保证 $100 \leq H_i \leq 199$ 且 H_i 互不相同。

3.2 编程题 2

- **试题名称**：数字替换
- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.2.1 题目描述

Alice 不喜欢数字 4，但觉得数字 8 寓意好，她想把数中的 4 全都替换成 8，若数中不含 4 则无需修改，你能帮帮她吗?

3.2.2 输入格式

输入一行，包含一个整数 A，表示替换前的数。

3.2.3 输出格式

输出一行，包含一个整数 B，表示替换后的数。

3.2.4 样例

3.2.4.1 输入样例 #1

```
8459045
```

3.2.4.2 输出样例 #1

```
8859085
```

3.2.4.3 输入样例 #2

```
123
```

3.2.4.4 输出样例 #2

```
123
```

3.2.4.5 样例解释

对于样例1，输入 8459045 中有两个 4，都将其替换为了 8，得到 8859085。

对于样例2，输入 123 中不包含 4，无需修改输入数字，输出 123。

3.2.5 数据范围

$0 \leq A \leq 10^8$。