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

C++ 二级

2025 年 03 月

1 单选题（每题 2 分，共 30 分）

第 1 题 2025年春节有两件轰动全球的事件，一个是DeepSeek横空出世，另一个是贺岁片《哪吒2》票房惊人，入了全球票房榜。下面关于DeepSeek与《哪吒2》的描述成立的是( )。

- A. 《哪吒2》是一款新型操作系统
- B. DeepSeek是深海钻探软件
- C. 《哪吒2》可以生成新的软件
- D. DeepSeek可以根据《哪吒2》的场景生成剧情脚本

第 2 题 对整型变量N，如果它能够同时被3和5整除，则输出 `N是含有至少两个质因数` 。如果用流程图来描述处理过程，则输出语句应该在哪种图形框中（ ）。

- A. 圆形框
- B. 椭圆形框
- C. 平行四边形框
- D. 菱形框

第 3 题 下面C++代码执行，其输出是( )。

```
int a=3, b = 4;
a == b;
b == a;
cout << a << ' ' << b << endl;
```

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

第 4 题 求三色彩球的颜色。有数量无限的红(Red)绿(Green)蓝(Blue)三种彩球排成一行，每组先为5个红色球，随后3个绿色，最后为2个蓝色。每个球都有编号，从左到右依次为1,2,3……。输入整数代表编号，求该编号球的颜色。下面是C++代码是实现，正确说法是( )。

```cpp
int N, remainder;
cin >> N;
remainder = N % 10; // remainder变量保存余数

if((1 <= remainder) && (remainder<= 5))
    cout << "Red";
else if ((6 <= remainder) &&  (remainder <= 8))
    cout << "Green";
else if ((remainder == 9) || (remainder == 0))
    cout << "Blue";
```

- [ ] **A.** 将 `else if ((remainder == 9) || (remainder == 0))` 修改为 `else` 效果相同
- [ ] **B.** 将 `((1 <= remainder) && (remainder<= 5))` 修改为 `(remainder <= 5)` 效果相同
- [ ] **C.** `else if ((6 <= remainder) &&  (remainder <= 8))` 写法错误，应修改为 `else if (6 <= remainder <= 8)`
- [ ] **D.** 根据题意 `remainder = N % 10` 应修改为 `remainder = N / 10`

第 5 题 下面C++代码执行后其输出是(　　)。

```cpp
int tnt = 0;
for (int i = 0; i < 10;i ++)
    if (i % 3)
        tnt += 1;
    else
        tnt += 2;
cout << tnt;
```

- [ ] **A.** 18
- [ ] **B.** 17
- [ ] **C.** 16
- [ ] **D.** 14

第 6 题 下面C++代码执行后输出是(　)。

```cpp
int i;
for (i = 10; i > 0; i -= 2)
    break;
cout << i;
```

- [ ] **A.** 10
- [ ] **B.** 8
- [ ] **C.** 0
- [ ] **D.** 因为循环执行时会执行break语句而终止循环，所以i的值不确定

第 7 题 下面C++代码执行后输出是（　　）。

```
int i;
for (i =0; i < 10; i++){
    if (i % 3 == 0)
        continue;
    cout << "0" << "#";
}
if (i >= 10)
    cout << "1" << "#";
```

- [ ] **A.** `0#0#0#0#0#0#`
- [ ] **B.** `0#0#0#0#0#0#0#1#`
- [ ] **C.** `0#0#0#0#1#`
- [ ] **D.** `0#0#0#0#0#0#1#`

第 8 题 下面C++代码执行后的输出是（　）。

```
int i,j;
for (i = 0; i < 5; i++)
    for (j = i; j > 0; j -= 1)
        printf("%d-",j);
```

- [ ] **A.** `1-2-1-3-2-1-4-3-2-1-`
- [ ] **B.** `1-2-1-3-2-1-4-3-2-1`
- [ ] **C.** `0-0-1-0-1-2-0-1-2-3-`
- [ ] **D.** `0-0-1-0-1-2-0-1-2-3`

第 9 题 下面C++代码执行后，将输出能被2整除且除以7余数为2的数。下列选项不能实现的是（　　）。

```
for (int i = 0; i < 100; i++)
    if ____________________
        cout << i << "  ";
```

- [ ] **A.** `((i % 2 == 0) && (i % 7 == 2))`
- [ ] **B.** `((!(i % 2)) && (i % 7 == 2))`
- [ ] **C.** `((!(i % 2)) && (!(i % 7)))`
- [ ] **D.** `((i % 2 != 1) && (i % 7 == 2))`

第 10 题 下面C++代码用于求1到N之间正整数中含有3的数的个数，比如 `123` 和 `32` 都是符合条件的数。则前后两处横线应填入代码分别是（　）。

```
int i,j;
int cnt = 0, N;
cout << "请输入正整数N: ";
cin >> N;
for (i = 1; (j=i) < N; i++)
    while (j != 0)
        if (j % 10 == 3){
            cnt +=1;
            __________;
        }
        else
            __________;
cout << cnt << "  ";
```

- [ ] **A.**

```
continue
j /= 10
```

- [ ] **B.**

```
break
j /= 10
```

- [ ] **C.**

```
continue
j %= 10
```

- [ ] **D.**

```
break
j %= 10
```

第 11 题 在数学中N!表示N的阶乘，即1到N的乘积，如 `3!=1*2*3` ，且 `0! = 1` 。下面的两段C++代码用于求1到N的阶乘之和，如N为3，则结果是9（ `1!+2!+3!` 的值）。选项中的说法正确的是（ ）。

```
// 实现1
int i,N;
cin >> N;
int tnt = 0, last = 1;
for (i = 1; i < N + 1; i++){
    last *= i;
    tnt += last;
}
cout << tnt << endl;
```

```cpp
// 实现2
int i,N;
cin >> N;
int tnt = 0, tmp;
for (i = 1; i < N + 1; i++){
    tmp = 1;
    for (int j = 1; j < i + 1; j++)
        tmp *= j;
    tnt += tmp;
}
cout << tnt << endl;
```

- [ ] **A.** 虽然实现1的代码短小，但效率并不高
- [ ] **B.** 实现2的代码效率更高，且更易于理解
- [ ] **C.** 实现1因为应用了前项计算结果，计算量更小，因此效率高
- [ ] **D.** 两种实现，效率几乎一致

第 12 题 哥德巴赫猜想是指大于2的偶数都可以分解为两个质数之和，下面的代码用于验证4-1000之内的偶数能否分解为两个质数之和。下面C++代码中假设isPrime()是已经定义好用于判断正整数N是否为质数，返回bool值。对该段代码，错误的说法是（ ）。

```cpp
for (i = 4; i < 1000; i += 2)
    for (j = 2; j < i; j++)
        if (isPrime(j) && isPrime(i-j)){
            printf("%d=%d+%d\n", i, j, i-j);
            break;
        }
```

- [ ] **A.** 将代码 `isPrime(j) && isPrime(i-j)` 修改为 `isPrime(j) == true && isPrime(i-j) == true` 效果相同
- [ ] **B.** 代码执行后，输出的一对质数，一定是小的数在前
- [ ] **C.** 即便将外层循环中i的上界1000修改为很大的整数，也不能说从数学上证明了哥德巴赫猜想
- [ ] **D.** 根据题意，break语句应该移到if语句块之外

第 13 题 已知C++代码和执行后的期望输出如下，相关说法正确的是（ ）。

```cpp
int i,j;
int last, N;

cout << "请输入层数N: ";
cin >> N;

last = 1;
for (i = 1; i < N; i++){
    for (j = 1; j < i + 1; j++){ // L1
        if (last > 9)
            last = 1;

        cout << last << " ";
```

```
        last += 1;
    }
    printf("\n");
}
```

```
请输入层数N：10
1
2 3
4 5 6
7 8 9 1
2 3 4 5 6
7 8 9 1 2 3
4 5 6 7 8 9 1
2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
```

- [ ] **A.** 倒数第二行的 `printf("\n")` 有错，应该修改为 `cout << endl;` ，printf( )函数不能输出换行
- [ ] **B.** `last += 1` 修改为 `last = last + 1` 执行效果相同
- [ ] **C.** 代码中L1标记行中的 `j < i + 1` 应修改为 `j < i`
- [ ] **D.** 外层for循环前的 `last = 1` 修改为 `last = 0` 执行效果相同

第 14 题 在C++中，（ ）最适合填入横线处连续5次正确生成1到10之间的随机整数?

```
for(int i=0; i<5; i++)
    ____________________;
```

- [ ] **A.** `rand( ) % 11`
- [ ] **B.** `rand( ) % 10`
- [ ] **C.** `rand( ) % 10 + 1`
- [ ] **D.** `rand() % 9 + 1`

第 15 题 在C++中，如果 `a` 和 `b` 均为 `float` 类型的变量，那么二者如果相差足够小（比如0.000001），就可以视作相等。比如 `2.2345676` 和 `2.2345677` 就可以视作相等。下列哪个表达式能用来正确判断“a等于b”( )。

- [ ] **A.** `((b-a) < 0.000001 )`
- [ ] **B.** `((b-a) <= 0.000001 )`
- [ ] **C.** `(abs(b-a) <= 0.000001 )`
- [ ] **D.** `(sqrt(b-a) <= 0.000001 )`

2 判断题（每题 2 分，共 20 分）

第 1 题 C++、Python都是高级编程语言，它们的每条语句最终都要通过机器指令来完成。（ ）

第 2 题 在C++代码中，假设N为正整数，则 `N - N / 10 * 10` 与 `N % 10` 都将获得N的个位数。( )

第 3 题 C++语句 `cout << ((10 <= N <= 12)? "true":"false")` 中，假设整型变量N为12，则其输出为 `true` 。原因是执行 `10 <= N` 后其值为 `true` ， `true` 与 12 相比仍然是 `true` 。(　　)

第 4 题 C++表达式 `(sqrt(N) * sqrt(N)) == N` 中的N如果为正整数，则表达式的值为true，相当于开平方后平方是本身。（ ）

第 5 题 下面C++执行后将输出 `3*2=6` 。(　　)

```
int a=2, b = 3;
a=a-b;
b=a+b;
a=b-a;
printf("%d*%d=%d\n", a, b, a*b);
```

第 6 题 下面C++代码执行后将输出10。（ ）

```
int i;
for (i = 0; i < 10; i++)
    continue;
cout << i << endl;
```

第 7 题 下面C++代码执行后将输出1。（ ）

```
int i;
for (i = 1; i < 10; i++){
    break;
    continue;
}
cout << i << endl;
```

第 8 题 下面的C++代码执行后将输出10行"OK"。（ ）

```
for (int i = 0; i < 5; i++)
    for(int j = 0; j < i; j++)
        printf("OK\n");
```

第 9 题 将下面C++代码中的for循环中的 `i = 1` 调整为 `i = 0` 的输出结果相同。（ ）

```
int tnt = 0;
for (int i = 1; i < 5; i++) // i=1
    tnt += i;
cout << tnt;
```

第 10 题 下面C++代码执行后将输出 `0123` 。（ ）

```
for (i = 0; i < 5; i++)
    for (i = 0; i < i; i++)
            continue;
    printf("%d\n", i);
```

3　编程题（每题 25 分，共 50 分）

3.1　编程题 1

- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.1.1　等差矩阵

3.1.2　题目描述

小 A 想构造一个 n 行 m 列的矩阵，使得矩阵的每一行与每一列均是等差数列。小 A 发现，在矩阵的第 i 行第 j 列填入整数 $i \times j$，得到的矩阵能满足要求。你能帮小 A 输出这个矩阵吗?

3.1.3　输入格式

一行，两个正整数 n, m。

3.1.4　输出格式

共 n 行，每行 m 个由空格分割的整数，表示小 A 需要构造的矩阵。

3.1.5　样例

3.1.5.1　输入样例 1

```
3 4
```

3.1.5.2　输出样例 1

```
1 2 3 4
2 4 6 8
3 6 9 12
```

3.1.6　数据范围

对于所有测试点，保证 $1 \leq n \leq 50$，$1 \leq m \leq 50$。

3.2 编程题 2

- **试题名称**：时间跨越
- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.2.8 题面描述

假设现在是 y 年 m 月 d 日 h 时而 k 小时后是 y' 年 m' 月 d' 日 h' 时，对于给定的 y, m, d, h, k，小杨想请你帮他计算出对应的 y', m', d', h' 是多少。

3.2.9 输入格式

输入包含五行，每行一个正整数，分别代表 y, m, d, h, k。

3.2.10 输出格式

输出四个正整数，代表 y', m', d', h' 。

3.2.11 样例

```
2008
2
28
23
1
```

```
2008 2 29 0
```

3.2.12 数据范围

对于全部数据，保证有 $2000 \leq y \leq 3000, 1 \leq m \leq 12, 1 \leq d \leq 31, 0 \leq h \leq 23, 1 \leq k \leq 24$。数据保证为合法时间。

3.2.13 提示

闰年判断规则

- 普通闰年：年份能被 4 整除，但不能被 100 整除。
- 世纪闰年：年份能被 400 整除。

满足以上任意一条规则的年份就是闰年，否则是平年。