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

C++ 二级

2025 年 09 月

1 单选题（每题 2 分，共 30 分）

第 1 题 人工智能现在非常火，小杨就想多了解一下，其中就经常听人提到“大模型”。那么请问这里说的“大模型”最贴切是指（ ）。

- A. 大电脑模型
- B. 大规模智能
- C. 智能的单位
- D. 大语言模型

第 2 题 在TCP协议中，完成连接建立需要通过（ ）握手。

- A. 一次
- B. 二次
- C. 三次
- D. 四次

第 3 题 下面的C++代码用于输入姓名，然后输出姓名，正确的说法是(　　)。

```
string XingMing;
cout << "请输入您的姓名：";
cin >> XingMing;
cout << XingMing;
```

- A. `XingMing` 是汉语拼音，不能作为变量名称
- B. 可以将 `XingMing` 改为 `Xing Ming`
- C. 可以将 `XingMing` 改为 `xingming`
- D. 可以将 `XingMing` 改为 `Xing-Ming`

第 4 题 下面C++代码用于获得正整数N的第M位数，如N等于1234，M等于2，则输出 `3` 。此题假设M的值大于等于1且小于等于N的位数。横线处应填入的代码是(　　)。

```
int N, M, div=1;
cout << "请输入一个正整数: ";
cin >> N;
cout <<"请输入从右到左取第几位数: ";
cin >> M;

for (int i =0; i < (M - 1); i++) div *= 10;

cout << (_____________);
```

- [ ] **A.** `N % div / 10`
- [ ] **B.** `N / div / 10`
- [ ] **C.** `N % div % 10`
- [ ] **D.** `N / div % 10`

第 5 题 下面C++代码执行，其输出是(　　)。

```
a, b = 3, 4;
c = a == b;
cout << a << ' ' << b << ' ' << c;
```

- [ ] **A.** `3 4 0`
- [ ] **B.** `3 3 3`
- [ ] **C.** `4 4 4`
- [ ] **D.** 以上都不对

第 6 题 某种编号的规则是“XX-Y”，其中XX从00到11，Y从0到9。第1个编号是00-0，第2个编号是01-1，…，第12个编号11-1，第13个编号00-2，即其编码规则是XX和Y同时增1，到XX到11时下一个变为00，Y到9时，下一个变为0。下面的C++代码用于生成第N个编号，横线处应填上的代码是(　　)。

```
cout << "请输入编号位置: ";
cin >> N;

part1 = N % _____;
part2 = N % _____;

if (part1 < 10)
    printf("0%d-%d\n", part1, part2);
else
    printf("%d-%d\n", part1, part2);
```

- [ ] **A.** `12 10`
- [ ] **B.** `10 10`
- [ ] **C.** `11 9`
- [ ] **D.** `9 9`

第 7 题 下面的C++代码执行后其输出是(　)。

```
cnt = 0;
for (int i = -10; i < 10; i++)
    for (int j =0; j < i; j++)
       cnt += 1;
cout << cnt;
```

- [ ] **A.** `145`

- [ ] **B.** `125`
- [ ] **C.** `55`
- [ ] **D.** `45`

第 8 题 下面C++代码执行后其输出是(　　)。

```
for (i =1; i < 12; i++){
    if (i % 2 == 0)
        continue;

    for (j = 0; j < i; j++)
        if (i * j % 2)
            break;
}
if(i>= 12)
    cout << (i * j);
```

- [ ] **A.** `110`
- [ ] **B.** `12`
- [ ] **C.** 不确定
- [ ] **D.** 无输出

第 9 题 阅读下面的C++代码，其中变量都是整型，则说法正确的是( )。

```
cin >>a >> b;

while (b != 0){
    remainder = a % b;
    a = b;
    b = remainder;
}
cout << a;
```

- [ ] **A.** `b` 不能为 `0`，因为 `a % b` 将导致错误
- [ ] **B.** `a` 必须小于 `b`，否则 `a % b` 将导致错误
- [ ] **C.** `a` 和 `b` 都必须为正整数，否则 `a % b` 将导致错误
- [ ] **D.** 如果 `a` 输入为 `0`，则不管 `b` 的输入值是什么，输出值的绝对值都是 `abs(b)`

第 10 题 下面C++代码执行后输出是（　）。

```
num = 0;
while (num <= 5){
    num += 1;
    if (num == 3)
        continue;
    printf("%d#", num);
}
```

- [ ] **A.** `1#2#4#5#6#`
- [ ] **B.** `1#2#4#5#6`
- [ ] **C.** `1#2#3#4#5#6#`
- [ ] **D.** `1#2#3#4#5#6`

第 11 题 下面C++代码用于记录多个输入数中的最大数和最小数（输入 `-999` 则输入结束），相关说法错误的是（）。

```
cin >> now_num;
min_num = max_num = now_num;

while (now_num != -999){
    if (max_num < now_num)
        max_num = now_num;

    if (min_num > now_num)
        min_num = now_num;

    cin >> now_num;
}
cout << min_num << ' ' << max_num;
```

- [ ] **A.** 程序运行时如果第一个数输入 `-999`，则输出将是 `-999 -999`
- [ ] **B.** 程序输入过程中，如果输入的第一个数不是 `-999`，则如果待输入的数据中没有 `-999`，则程序能求出已输入整数中的最大数和最小数
- [ ] **C.** 如果用于输入考试成绩，即成绩中不可能有 `-999`，则程序能求出已输入成绩中的最高成绩和最低成绩
- [ ] **D.** 可以将 `cin >> now_num;` 移动到 `while (now_num != -999) {` 下面，结果不变

第 12 题 下面C++代码执行后输出与 5 有关数的数量。“与 5 有关的数”定义为含有 5 或者能被 5 整除的数。相关说法正确的是（　）。

```
cnt = 0;
for (i = 1; i < 1000; i++){
    if (i % 5 == 0){
        cnt += 1;
        continue;
    }
    j = i;
    while (j > 0)
        if (j % 10 == 5){
            cnt += 1;
            break;
        }
        else
            j /= 10;
}
cout << cnt;
```

- [ ] **A.** 删除代码中 `continue` 不影响程序执行结果
- [ ] **B.** 删除 `j = i` 并将 `while` 循环内的j修改为 `i`，不影响程序执行结果
- [ ] **C.** 代码中 `break` 修改为 `j = 0`，不影响程序执行结果
- [ ] **D.** 将 `while (j > 0)` 修正为 `while (j >= 0)` 不影响程序执行的结果

第 13 题 下面C++代码实现输出如下图形，应该在横线处填入的代码是（　）。

```
请输入层数：10
         1
        23
       456
      7891
     23456
    789123
   4567891
  23456789
 123456789
1234567891
```

```
cout << "请输入层数: ";
cin >> N;

K = 1;

for (i = 1; i < N + 1; i++){
    for (int _ = 1; _ < ___________; _++)
        cout << " ";

    for (int _ = 1; _ < ________; _++){
        cout << K;

        K += 1;
        if (K == 10)
            K = 1;
    }
    cout << '\n';
}
```

- [ ] **A.**

```
N - i + 1
i + 1
```

- [ ] **B.**

```
N - i
i
```

- [ ] **C.**

```
N
i
```

- [ ] **D.**

```
N - i
i + 1
```

第 14 题 下面C++代码执行，其输出是(　)。

```
int a=9, b=27;
a = 'a' + 'b';
b = 'a' - 'b';
a = a - b;
cout << a << ' ' << b << endl;
```

- [ ] **A.** `196 -1`
- [ ] **B.** `27 9`
- [ ] **C.** `98 97`
- [ ] **D.** 不确定

第 15 题 有个无限长的链，由3种外形相同但材质不同的环链成。3种环的重量分别是3、4、6克，相同材质的多个环每12克一组，分别记为G3、G4、G6。链依次G3、G4、G6、G3、G4、G6、…。同时对链上所有环从头依次编号1、2、3、4……。输入正整数代表环编号，求该编号前所有环（不含该环本身）的重量。下面是C++代码是实现，正确说法是(　)。

```
i int N, G, R;
int wc = 0;

cin >> N;  // 输入正整数
```

```
G = (N - 1) / 9; // L1
R = (N - 1) % 9; // L2: 保存余数
wc += 36*G;

if((1 <= R) && (R <= 4))
    wc += 3*R; // L3
else if ((5 <= R) &&  (R <= 7))
    wc += 4*R; // L4
else if(R == 8)
    wc += 6*(R - 1); // L5
cout << wc << endl;
```

- ☐ **A.** 必须同时修改 `L1` 和 `L2` 代码行才能实现功能
- ☐ **B.** 必须同时修改 `L3` 和 `L4` 代码行才能实现功能
- ☐ **C.** 必须同时修改 `L3` 和 `L5` 代码行才能实现功能
- ☐ **D.** 其他说法都不对

2 判断题（每题 2 分，共 20 分）

第 1 题 在集成开发环境里调试程序时，要注意不能修改源程序，因为如果修改，就要终止调试、关闭该文件并重新打开，才能再次开始调试。（ ）

第 2 题 在C++代码中，假设N为正整数且大于100，则 `N / 100` 将舍弃个位和十位，如N为1234则 `cout << (N / 100)` 将输出 `12` 。如果N小于100，则其值为 `0` 。(　　)

第 3 题 下列C++代码执行后将输出 `1` ，因为 `a` 确实小于 20 和 10。(　　)

```
a = 5;
cout << (a < 10 and 20);
```

第 4 题 下面的C++代码中变量都是整型，则执行后将输出 `1` 。（ ）

```
x, y, z = 5, 10, 15;
result = x < y < z;
cout << result;
```

第 5 题 下面C++代码执行时如输入 `99.99` ，将输出 及格 两个汉字。（ ）

```
int score;
cout << "请输入学生成绩：";
cin >> score;
if (score >= 60)
    printf("及格");
else
    printf("不及格");
```

第 6 题 执行下面的C++代码时输入 `123` ，则输出是 `DCB` 。(　　)

```
int a;
cin >> a;
while(a){
    cout << 'A'+a%10;
    a /= 10;
}
```

第 7 题 下面的C++代码执行后将输出 `+#+#3#` 。（ ）

```
int i;
for (i = 0; i < 3; i++){
    if (i == 2)
        continue;
    printf("+#");
}
cout << i << '#';
```

第 8 题 下列C++代码用于求斐波那契数列，即第1个数0，第2个数1，从第三个数开始，是前两个数之和。如果输入的值为大于1的正整数，该代码能实现。（ ）

```
cin >> n;
a = 0, b = 1;
for (int j = 0; j < n; j++){
    cout << a << " ";
    b = b + a;
    a = b - a;
}
```

第 9 题 下面的C++不能实现如下输出，但如果将L1标记的 `cout << 0` 行移动 `if` 块外面，或者说移动到 `L2` 标记行，则可以。（ ）

```
请输入矩阵大小n: 9
100000000
020000000
003000000
000400000
000050000
000006000
000000700
000000080
000000009
```

```
int n, i, j;
cout << "请输入矩阵大小n: ";
cin >> n;

for (i = 0; i < n; i++){
    for  (j = 0; j < n; j++){
        if (i == j){
            cout << i + 1;
            continue;
            cout << 0; // L1
        }
        // L2
    }
    printf("\n");
}
```

第 10 题 C++代码 `cout << ('5'+4);` 执行后的输出为 `9` 。（ ）

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- **试题名称**：优美的数字
- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.1.1 题目描述

如果一个正整数在十进制下的所有数位都相同，小 A 就会觉得这个正整数很优美。例如，正整数 6 的数位都是 6，所以 6 是优美的。正整数 99 的数位都是 9，所以 99 是优美的。正整数 123 的数位不都相同，所以 123 并不优美。

小 A 想知道不超过 n 的正整数中有多少优美的数字。你能帮他数一数吗?

3.1.2 输入格式

一行，一个正整数 n。

3.1.3 输出格式

一行，一个正整数，表示不超过 n 的优美正整数的数量。

3.1.4 样例

3.1.4.1 输入样例 1

```
6
```

3.1.4.2 输出样例 1

```
6
```

3.1.4.3 输入样例 2

```
2025
```

3.1.4.4 输出样例 2

```
28
```

3.1.5 数据范围

对于所有测试点，保证 $1 \leq n \leq 2025$。

3.2 编程题 2

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

3.2.1 题目描述

小 A 想绘制一个菱形。具体来说，需要绘制的菱形是一个 n 行 n 列的字符画，n 是一个大于 1 的奇数。菱形的四个顶点依次位于第 1 行、第 1 列、第 n 行、第 n 列的正中间，使用 `#` 绘制。相邻顶点之间也用 `#` 连接。其余位置都是 `.`。例如，一个 5 行 5 列的菱形字符画是这样的：

```
..#..
.#.#.
#...#
.#.#.
..#..
```

给定 n，请你帮小 A 绘制对应的菱形。

3.2.2 输入格式

一行，一个正整数 n。

3.2.3 输出格式

输出共 n 行，表示对应的菱形。

3.2.4 样例

3.2.4.1 输入样例 1

```
3
```

3.2.4.2 输出样例 1

```
.#.
#.#
.#.
```

3.2.4.3 输入样例 2

```
9
```

3.2.4.4 输出样例 2

```
....#....
...#.#...
..#...#..
.#.....#.
#.......#
.#.....#.
..#...#..
...#.#...
....#....
```

3.2.5 数据范围

对于所有测试点，保证 $3 \leq n \leq 29$ 并且 n 为奇数。