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

C++ 一级

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 小杨这学期刚开学就选修了一门编程课，然后就想编写程序来计算1到10001之间的所有偶数的和。他希望程序采用简单累加整数的方法，同时希望源程序尽可能清晰、简洁，则下面关于实现累加时采用的主要控制结构的哪个说法最不合适（ ）。

- A. 使用循环结构
- B. 使用循环和分支的组合
- C. 仅使用顺序结构
- D. 不使用分支结构

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 下列C++代码中a和b都是整型变量，执行后，其结果是( )。

```
a = 13;
b = 5;
cout << a / b << a %// b << a % b;
a*b;
```

- A. `2 3`

- [ ] **B.** `23`
- [ ] **C.** `20`
- [ ] **D.** 以上都不准确

第 5 题 C++表达式 `3 * 4 % 5 / 6` 的值是(　)。

- [ ] **A.** 10
- [ ] **B.** 5
- [ ] **C.** 2
- [ ] **D.** 0

第 6 题 下面的C++代码中变量N和M都是整型，则执行时如果先输入10并输入一个制表符后输入20并回车，其输出的数值是(　　)。

```
scanf("%d", &N);
scanf("%d", &M);
printf("{%d}",N+M);
```

- [ ] **A.** `{30}`
- [ ] **B.** `1020`
- [ ] **C.** `{N+M}`
- [ ] **D.** 不输出，继续等待输入

第 7 题 当前是9月，编写C++代码求N个月后的月份。横线处应填入的代码是（　）。

```
int N, M;
cin >> N;
M = _____________;
if (M == 0)
    printf("%d个月后12月", N);
else
    printf("%d个月后是%d月", N, M);
```

- [ ] **A.** `N % 12`
- [ ] **B.** `9 + N % 12`
- [ ] **C.** `(9 + N) / 12`
- [ ] **D.** `(9 + N) % 12`

第 8 题 下面C++代码执行后的输出是（　　）。

```
int n = 0;
for (int i =0; i < 100; i++)
    n += i % 2;
cout << n;
```

- [ ] **A.** `5050`
- [ ] **B.** `4950`
- [ ] **C.** `50`
- [ ] **D.** `49`

第 9 题 下面的C++代码执行后输出是（　　）。

```
int N = 0, i;
for (i = -100; i < 100; i++)
    N += i % 10;
cout << N;
```

- [ ] **A.** `900`
- [ ] **B.** `100`
- [ ] **C.** `0`
- [ ] **D.** `-100`

第 10 题 下面C++代码执行后输出是（ ）。

```
int i;
for(i = 1; i < 5; i++){
    if(i % 3 == 0)
        break;
    printf("%d#",i);
}
if(i > 5) printf("END\n");
```

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

第 11 题 下面的C++代码用于求N的镜面数（N的个位到最高位的各位数字依次反过来出现在数字中，但高位0将被忽略，不输出），如输入1234，则将输出 `4321`，又如输入120，则将输出 `21`，错误的选项是（ ）。

```
cout << "请输入个位数不为0的正整数: ";
cin >> N;
rst = 0; // 保存逆序结果
while (_____________){
    rst = rst * 10 + N % 10;
    N  = N / 10;
}
cout << rst << endl;
```

- [ ] **A.** `N != 0`
- [ ] **B.** `not (N == 0)`
- [ ] **C.** `N = 0`
- [ ] **D.** `N > 0`

第 12 题 下面C++代码用于交换两个正整数a和b的值，不能实现交换的代码是（ ）。

- [ ] **A.**

```
cout << "输入第一个正整数: ";
cin >> a;
cout <<"输入第二个正整数: ";
cin >> b;
temp = a;
a = b;
b = temp;
cout << "a=" << a << " b=" << b << endl;
```

- [ ] **B.**

```
cout << "输入第一个正整数: ";
cin >> a;
cout <<"输入第二个正整数: ";
cin >> b;
b = a - b;
a = a - b;
b = a + b;
cout << "a=" << a << " b=" << b << endl;
```

☐ **C.**

```
cout << "输入第一个正整数: ";
cin >> a;
cout <<"输入第二个正整数: ";

cin >> b;
a = a + b;
b = a - b;
a = a - b;
cout << "a=" << a << " b=" << b << endl;
```

☐ **D.**

```
cout << "输入第一个正整数: ";
cin >> a;
cout <<"输入第二个正整数: ";
cin >> b;
a, b = b, a;
cout << "a=" << a << " b=" << b << endl;
```

第 13 题 下面C++代码用于获得正整数N的第M位数，约定个位数为第1位，如N等于1234，M等于2，则输出3。假设M的值是大于等于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 14 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 15 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

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 执行C++表达式 `10 % 0.5` 将报错，因为 `0.5` 所在位置只能是整数。( )

第 3 题 下面C++代码执行后将输出 `9`。( )

```
for (i = 0; i < 10; i++)
    break;
cout << i;
```

第 4 题 下面C++代码执行后将输出 `55`。（ ）

```
n = 0;
for (int i = 0; i > -10; i--)
    n = n + i * -1;
cout << n;
```

第 5 题 将下面C++代码中的 `L1` 行的 `i = 0` 修改为 `i = 1`，其输出与当前代码输出相同。（ ）

```
cnt = 0;
for (int i = 0; i < 100; i++) // L1
    cnt += i;
cout << cnt
```

第 6 题 将下面C++代码中的 `i < 10` 修改为 `i <= 10`，其执行后输出相同。（ ）

```
int n, i;
n = i = 0;
while (i < 10){
    n += i;
    i += 1;
}
cout << n;
```

第 7 题 下面的C++代码执行后将输出 `45`。（ ）

```
int n, i;
n = i = 0;
while (i < 10){
    i += 1;
    n += i;
}
cout << n;
```

第 8 题 执行C++代码 `cout << (12 + 12.12)` 将报错，因为 `12` 是 `int` 类型，而 `12.12` 是 `floa t`类型，不同类型不能直接运算。（ ）

第 9 题 下面C++代码执行时将导致无限循环（也称死循环）。（ ）

```
int count = 0;
while (count < 5){
    count += 1;
    if (count == 3)
        continue;
    cout << count << ' ';
}
```

第 10 题 下列C++代码用于求斐波那契数列，即第1个数为0，第2个数为1，从第三个数开始，依次是其前两个数之和。如果输入的值为大于1的正整数，该代码能实现所求。（ ）

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

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- **试题名称**：商店折扣
- **时间限制**：1.0 s
- **内存限制**：512.0 MB

3.1.1 题目描述

商店正在开展促销活动，给出了两种方案的折扣优惠。第一种方案是购物满 x 元减 y 元；第二种方案是直接打 n 折，也就是说价格变为原先的 $\frac{n}{10}$。这里的 x, y, n 均是正整数，并且 $1 \leq y < x$，$1 \leq n < 10$。

需要注意的是，第一种方案中满减优惠只能使用一次。例如购物满 10 元减 3 元时，若挑选了价格总和为 33 元的物品，只能减免 3 元，需要支付 30 元。

小明在商店挑选了价格总和为 p 元的物品，结账时只能使用一种优惠方案。小明最少需要支付多少钱呢?

3.1.2 输入格式

四行，四个正整数 x, y, n, p，含义见题目描述。

3.1.3 输出格式

一行，一个小数，表示小明最少需要支付多少钱，保留两位小数。

3.1.4 样例

3.1.4.1 输入样例 1

```
8
7
9
10
```

3.1.4.2 输出样例 1

```
3.00
```

3.1.4.3 输入样例 2

```
8
7
2
11
```

3.1.4.4 输出样例 2

```
2.20
```

3.1.5 数据范围

对于所有测试点，保证 $1 \leq y < x \leq 100$，$1 \leq n < 10$，$1 \leq p \leq 100$。

3.2 编程题 2

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

3.2.1 题目描述

金字塔由 n 层石块垒成。从塔底向上，每层依次需要 $n \times n, (n-1) \times (n-1), \ldots, 2 \times 2, 1 \times 1$ 块石块。请问搭建金字塔总共需要多少块石块?

3.2.2 输入格式

一行，一个正整数 n，表示金字塔的层数。

3.2.3 输出格式

一行，一个正整数，表示搭建金字塔所需的石块数量。

3.2.4 样例

3.2.4.1 输入样例 1

```
2
```

3.2.4.2 输出样例 1

```
5
```

3.2.4.3 输入样例 2

```
5
```

3.2.4.4 输出样例 2

```
55
```

3.2.5 数据范围

对于所有测试点，保证 $1 \leq n \leq 50$。