



C++ 二级

2026 年 03 月

1 单选题（每题 2 分，共 30 分）

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	B	D	D	C	B	A	A	C	C	A	B	C	B	A	A

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。如果将机器人视为一个计算机系统，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 下面代码用来找出输入的 N 个正整数中最大的一个。如果将代码段用流程图来表示，则 L1 标记的代码行应该使用的图形是 ()。

```
1 int N, max=0, val;  
2 cin >> N;  
3  
4 while(N){  
5     cin >> val;  
6     if(val > max) // L1  
7         max = val;  
8     N--;  
9 }  
10  
11 cout << max;
```

- 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++ 代码可以执行，有关说法正确的是()。

```
1 double PI = 3.1415926;  
2 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 下面选择项中，与C++表达式 $!(x > 5 \ \&\& \ y \leq 10)$ 等价的是()。

- A. $x \leq 5 \ \&\& \ y > 10$
- B. $x > 5 \ || \ y \leq 10$
- C. $x \leq 5 \ || \ y > 10$
- D. $! x > 5 \ \&\& \ ! y \leq 10$

第5题 某同学执行C++代码 `cout << ((0.1 + 0.2) == 0.3)` 时输出 0，下面最可能的原因是()。

- A. C++的 + 运算符在处理小数时存在bug
- B. 0.1、0.2 和 0.3 在计算机中无法用二进制浮点数精确表示，导致 0.1 + 0.2 的结果与 0.3 存在微小误差
- C. == 运算符不能用于比较浮点数，只能用于整数
- D. 因为 0.1 + 0.2 的数学结果不等于 0.3

第6题 下面的 C++ 代码段执行后其输出是()。

```
1 | tnt = 0;
2 | for (int i = 0; i < 5; i++){
3 |     for (int j = 0; j < i; j++){
4 |         tnt += 1;
5 |         cout << tnt << "#";
6 |     }
7 | cout << tnt;
```

- A. 0#1#3#6#10#10
- B. 1#2#3#4#5#6#7#8#9#10#10
- C. 10#10
- D. 10

第7题 下面的 C++ 代码执行之后的输出是()。

```
1 | for ( int i = -2; i < 2; i++)
2 |     if (not i % 3)
3 |         cout << i << "#";
```

- A. 0#
- B. -2#-1#1#
- C. -1#0#
- D. -2#0#1#

第8题 下面的 C++ 代码执行后其输出是()。

```

1 int cnt = 0, i, j;
2 for (i =1; i < 5; i++){
3     for (j =0; j < i; j++)
4         cout << j << "#";
5     break;
6 }
7 if(i >= 5)
8     cout << (i*j);

```

- A. 0#0#1#0#1#2#0#1#2#3#12
- B. 0#0#1#0#1#2#0#1#2#3#
- C. 0#
- D. 1#

第9题 下面 C++ 代码执行后其输出是()。

```

1 int count = 0;
2 for (int i = 1; i < 4; i++)
3     for (int j = 1; j < 5; j++){
4         if (j == 3)
5             continue;
6         if (i == 2)
7             break;
8         count += 1;
9     }
10 cout << (count);

```

- A. 2
- B. 4
- C. 6
- D. 8

第10题 下面4个选项中，与下面 C++ 代码段具有相同效果的是()。

```

1 i = 0;
2 while (i < 5){
3     cout << i;
4     i += 1;
5 }

```

- A.

```

1 for (i = 0; i < 5; i++)
2     cout << i;

```

- B.

```

1 for (i = 1; i < 5; i++)
2     cout << i;

```

- C.

```

1 for (i = 0; i < 6; i++)
2     cout << i;

```

D.

```
1 for (i = 1; i < 6; i++)
2   cout << i;
```

第 11 题 下面 C++ 代码执行后输出是 ()。

```
1 int n = 10;
2 while (n > 0){
3     n -= 1;
4     if (n % 3 == 0)
5         continue;
6     if (n == 5)
7         break;
8 }
9 cout << n;
```

A. 0

B. 5

C. 6

D. 7

第 12 题 下面 C++ 代码段执行后，其输出是 ()。

```
1 int i, j, cnt;
2 cnt = 0;
3 for (i = 0; i < 5; i++){
4     i = -i;
5     for (j = i; j < -i; j++)
6         cnt += 1;
7     i = -i;
8 }
9 cout << cnt;
```

A. 5

B. 15

C. 20

D. 30

第 13 题 某学校图书馆的借阅卡号由6位整数组成。前5位是顺序编号，第6位是校验码，用于防止输错。校验码规则如下：将前5位数字相加，然后除以10的余数，就是第6位数字。如卡号 123455 的前5位之和为 15，除以 10 的余数是5，故第6位为5。下面的C++代码段用于判断卡号是否正确，横线处应填入的代码是()。

```

1  cout << "请输入卡号: ";
2  cin >> N;
3  order_num = N / 10; // 获得前5位顺序号,假设录入一定为6位正整数
4  check_num = N % 10; // 获得最后一位
5
6  tnt = 0; // 保存前5位之和
7  for (i = 0; i < 5; i++){
8      -----;
9      order_num /= 10;
10 }
11
12 if (-----)
13     cout << "符合校验规则";
14 else
15     cout << "不符合校验规则";

```

A.

```

1  tnt += order_num / 10
2  tnt / 10 == check_num

```

B.

```

1  tnt += order_num % 10
2  tnt % 10 == check_num

```

C.

```

1  tnt = order_num / 10 + tnt
2  tnt % 10 == check_num

```

D.

```

1  tnt = order_num % 10
2  tnt / 10 == check_num

```

第 14 题 下面的 C++ 代码段正常执行后其输出的数字图形是 ()。

```

1  for (i = 1; i < 5; i++){
2      for (j = 1; j < i+1; j++)
3          cout << j;
4      cout << endl;
5  }

```

A.

```

1  1
2  12
3  123
4  1234

```

B.

```

1  1
2  22
3  333
4  4444

```

C.

```
1 | 1
2 | 21
3 | 321
4 | 4321
```

D.

```
1 | 4
2 | 34
3 | 234
4 | 1234
```

第 15 题 某学校举办“校园演讲比赛”，每位选手由8位评委打分（分数为 0~10 的整数），且每位评委必须打分。计分规则：若至少有5位评委给出大于等于6分，则成绩有效，最终得分为所有8位评委的总分；如给出低于6分的评委数量超过5位，则记为0分。以下核心程序段依次输入8个分数，并计算最终得分。横线处应填入（ ）。

```
1 total_score = 0; // 所有分数之和
2 high_count = 0; // ≥6分的评委数量
3
4 for (i = 0; i < 8; i++){
5     cout << "请输入评委分数: ";
6     cin >> score;
7     _____;
8     if (score >= 6)
9         _____;
10 }
11
12 if (high_count >= 5)
13     cout << total_score;
14 else
15     cout << 0;
```

A.

```
1 total_score += score
2 high_count += 1
```

B.

```
1 total_score += score
2 high_count += score
```

C.

```
1 high_count += 1
2 total_score += score
```

D.

```
1 total_score *= score
2 high_count *= 1
```

2 判断题（每题 2 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	√	×	×	√	√	√	√	×	×	√

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++ 语句 `cout << ('4' % '2' == '2' * '2' % 2)`; 执行后的输出是1。()

第3题 下面的 C++ 代码段执行时将报错, 因为10是整数类型, `a` 是布尔类型。由于类型不同, 不能进行加法运算。()

```
1 bool a = true;
2 cout << (10 + a);
```

第4题 下面C++代码段执行后将输出 0-3-6-9-。()

```
1 for (int i = 0; i < 10; i++){
2     if (i % 3)
3         continue;
4     cout << i << "-";
5 }
```

第5题 执行下面的 C++ 代码段, 如果N是基本数据类型, 则语句 `cout << (N)`; 将被执行 0 次或无数次 (即死循环)。()

```
1 cin >> N;
2 while (N)
3     cout << (N);
```

第6题 下面的 C++ 代码段可正常执行, 删除 `continue` 不影响执行效果。()

```
1 for (i = 0; i < 10; i++){
2     i += 1;
3     continue;
4 }
5 cout << (i);
```

第7题 下面的 C++ 代码段用于计算1到N之间且包含N的所有数字中含有的3的个数, 如1到40含有14个3, 而1到20则只含有2个3。如果将 `while (i != 0)` 改为 `while (abs(i))`, 则执行结果相同。()

```
1 cout << "请输入正整数N: ";
2 cin >> N;
3 cnt = 0; // 保存3的个数
4 for (k = 1; k < N + 1; k++){
5     i = k;
6     while (i != 0){
7         if (i % 10 == 3)
8             cnt += 1;
9         i /= 10;
10    }
11 }
12 cout << cnt;
```

第8题 下面的 C++ 代码段执行将不会有输出, 因为内层循环 `j` 总是 0 开始, `i * j % 10 == 0` 将会被满足, 执行 `break`, 故而 `i` 小于 10, 不会满足 `if` 判断条件。()

```
1 for (i = 1; i < 10; i++)
2     for (j = 0; j < i; j++)
3         if (i * j % 10 == 0)
4             break;
5 if(i >= 10)
6     cout << (i*j);
```

第9题 下列 C++ 代码执行后将输出 1#4#9#16#16。()

```

1 cnt = 0;
2 for (i = 1; i < 5; i++){
3     for (j = 1; j < i + 1; j++){
4         if (i * j % 10 == 0)
5             break;
6         if (j >= i + 1)
7             cout << (i * j) << "#";
8     }
9     if(i >= 5)
10        cout << (i * j);

```

第 10 题 下面 C++ 代码执行后输出如左图所示，将 "%d" 修改为 "%3d" 即可实现右图输出。（ ）

```

1  /*
2     // 左图                // 右图
3     1 2 3 4 5 6 7 8 9      1  2  3  4  5  6  7  8  9
4     2 4 6 8 10 12 14 16 18  2  4  6  8 10 12 14 16 18
5     3 6 9 12 15 18 21 24 27  3  6  9 12 15 18 21 24 27
6     4 8 12 16 20 24 28 32 36  4  8 12 16 20 24 28 32 36
7     5 10 15 20 25 30 35 40 45  5 10 15 20 25 30 35 40 45
8     6 12 18 24 30 36 42 48 54  6 12 18 24 30 36 42 48 54
9     7 14 21 28 35 42 49 56 63  7 14 21 28 35 42 49 56 63
10    8 16 24 32 40 48 56 64 72  8 16 24 32 40 48 56 64 72
11    9 18 27 36 45 54 63 72 81  9 18 27 36 45 54 63 72 81
12  */
13
14  for (i = 1; i < 10; i++){
15      for (j = 1; j < 10; j++){
16          printf(" %d", i*j);
17          printf("\n");
18      }

```

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

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

3.1.1 题目描述

对于正整数 n ，如果 n 的所有数位中包含恰好 3 个 2，Alice 会认为这个正整数是美丽的。例如，正整数 24122 中包含 3 个 2，所以 24122 是美丽的；正整数 132 中包含 1 个 2，所以 132 不是美丽的；正整数 212322 中包含 4 个 2，所以 212322 不是美丽的。

Alice 想知道正整数 L 到正整数 R （包括 L 和 R ）中有多少美丽的正整数，你能帮她数一数吗？

3.1.2 输入格式

输入共 2 行，第一行为正整数 L ，第二行为正整数 R 。

3.1.3 输出格式

输出一行，包含一个整数 C ，表示 L 到 R 中 C 美丽数的数量。

3.1.4 样例

3.1.4.1 输入样例

```
1 | 2221
2 | 2223
```

3.1.4.2 输出样例

```
1 | 2
```

3.1.5 样例解释

2221 到 2223 中，2221 与 2223 是美丽的，2222 不是美丽的。

3.1.6 数据范围

保证 $1 \leq L \leq R \leq 10^6$ 。

3.1.7 参考程序

```
1 | #include <iostream>
2 |
3 | using namespace std;
4 |
5 | int main() {
6 |     int l, r, ans = 0;
7 |     cin >> l >> r;
8 |     for(int i = l; i <= r; i++) {
9 |         int c = 0, t = i;
10 |         while(t) {
11 |             if(t % 10 == 2)
12 |                 c++;
13 |             t /= 10;
14 |         }
15 |         if(c == 3)
16 |             ans++;
17 |     }
18 |     cout << ans;
19 |     return 0;
20 | }
```

3.2 编程题 2

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

3.2.1 题目描述

输入一个正整数 n ，你需要绘制一个 n 行 n 列的正方形，绘制规则如下：

- 正方形的四个顶点使用 + 绘制；
- 除顶点外，第 1 行与第 n 行使用 - 绘制；
- 除顶点外，第 1 列与第 n 列使用 | 绘制；
- 正方形内部使用 * 绘制。

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，表示对应的正方形。

3.2.4 样例

3.2.4.1 输入样例

```
1 | 5
```

3.2.4.2 输出样例

```
1 | +---+
2 | |***|
3 | |***|
4 | |***|
5 | +---+
```

3.2.5 数据范围

保证 $3 \leq n \leq 100$ 。

3.2.6 参考程序

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for(int i = 0; i < n; i++) {
9         for(int j = 0; j < n; j++) {
10            if(j == 0 || j == n - 1) {
11                if(i == 0 || i == n - 1)
12                    cout << '+';
13                else
14                    cout << '|';
15            } else {
16                if(i == 0 || i == n - 1)
17                    cout << '-';
18                else
19                    cout << '*';
20            }
21        }
22        if(i + 1 != n)
23            cout << endl;
24    }
25    return 0;
26 }
```