



# C++ 八级

2025 年 03 月

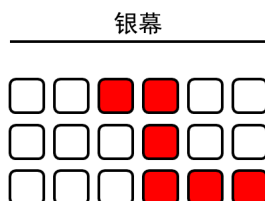
## 1 单选题（每题 2 分，共 30 分）

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

第 1 题 国家“以旧换新”政策仍在继续，小杨家决定在家里旧的冰箱、电视、洗衣机、微波炉中选两种换新。其中，冰箱有 4 种型号可选，电视有 6 种型号可选，洗衣机有 3 种型号可选，微波炉有 5 种型号可选。请问小杨家共有多少种换新的方案？（ ）。

- ☐ A. 18
- ☐ B. 119
- ☐ C. 238
- ☐ D. 360

第 2 题 小杨和 3 位朋友约好一起去看电影“哪吒 2”。打开购票软件，他们发现，已经没有同一排连续的四个座位了（图中每个方框代表一个座位，红色方框代表已经售出）。朋友们商量了一下，决定分为两组，每组两人在同一排的相邻两个座位，且两组之间至少有一对座位是前后相邻的。请问共有多少种购票方案？（ ）。



- ☐ A. 495
- ☐ B. 96
- ☐ C. 7
- ☐ D. 4

第 3 题 下面关于 C++ 类构造和析构函数的说法，错误的是（ ）。

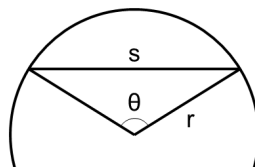
- ☐ A. 构造函数不能声明为虚函数。
- ☐ B. 析构函数必须声明为虚函数。
- ☐ C. 类的默认构造函数可以被声明为 private。
- ☐ D. 类的析构函数可以被声明为 private。

- ☐ A. 树是一种有向无环图，有向无环图都是一棵树。
- ☐ B. 如果把树看做有向图，每个节点指向其子节点，则该图是弱连通图。
- ☐ C.  $N$ 个顶点且连通的无向图，其最小生成树一定包含  $N - 1$  个条边。
- ☐ D.  $N + 1$  个顶点、 $N$  条边的有向图，一定不是强连通的。

第 5 题 从1到2025这2025个数中，包含数字5的个数（ ）。

- ☐ A. 600
- ☐ B. 601
- ☐ C. 602
- ☐ D. 603

第 6 题 已定义 `double` 类型的变量 `r` 和 `theta`，分别表示图中圆半径和圆心角。下列表达式中可以求出弦长 `s` 的是（ ）。



- ☐ A. `r * cos(theta)`
- ☐ B. `r * cos(theta / 2) * 2`
- ☐ C. `r * sin(theta)`
- ☐ D. `r * sin(theta / 2) * 2`

第 7 题  $N$  个节点的平衡二叉树的高为（ ）。

- ☐ A.  $\lfloor \log_2 N \rfloor$
- ☐ B.  $\lceil \log_2 N \rceil$
- ☐ C.  $\lfloor \log_2 N \rfloor + 1$
- ☐ D. 无法确定。

第 8 题 下列关于算法的说法，错误的是（ ）。

- ☐ A. 如果有足够的时间和空间，枚举法能解决一切有限的问题。
- ☐ B. 分治算法将原问题分为多个子问题进行求解，且分解出的子问题必须相互独立。
- ☐ C. 如果能找到合理的贪心原则，贪心算法往往能够比其他方法更快求解。
- ☐ D. 倍增法在搜索未知长度的有序数组时，通过动态倍增或减半步长，快速定位目标范围。

第 9 题 2025是个神奇的数字，因为它是由两个数20和25拼接而成，而且  $2025 = (20 + 25)^2$ 。小杨决定写个程序找找小于 $N$ 的正整数中共有多少这样神奇的数字。下面程序横线处应填入的是（ ）。

```
1  #include <string>
2  int count_miracle(int N) {
```

```

3   int cnt = 0;
4   for (int n = 1; n * n < N; n++) {
5       int n2 = n * n;
6       std::string s = std::to_string(n2);
7       for (int i = 1; i < s.length(); i++)
8           if (s[i] != '0') {
9               std::string sl = s.substr(0, i);
10              std::string sr = s.substr(i);
11              int n1 = std::stoi(sl);
12              int nr = std::stoi(sr);
13              if (_____) // 在此处填入选项
14                  cnt++;
15          }
16    }
17    return cnt;
18 }

```

☐ A.

1 | `n1 + nr == n`

☐ B.

1 | `n1 + nr == n2`

☐ C.

1 | `(n1 + nr) * (n1 + nr) == n`

☐ D.

1 | `(n1 + nr) ^ 2 == n2`

**第10题** 2025是个神奇的数字，因为它是由两个数20和25拼接而成，而且 $2025 = (20 + 25)^2$ 。小杨决定写个程序找找小于 $N$ 的正整数中共有多少这样神奇的数字。该函数的时间复杂度为（ ）。

```

1  #include <string>
2  int count_miracle(int N) {
3      int cnt = 0;
4      for (int n = 1; n * n < N; n++) {
5          int n2 = n * n;
6          std::string s = std::to_string(n2);
7          for (int i = 1; i < s.length(); i++)
8              if (s[i] != '0') {
9                  std::string sl = s.substr(0, i);
10                 std::string sr = s.substr(i);
11                 int n1 = std::stoi(sl);
12                 int nr = std::stoi(sr);
13                 if (_____) // 在此处填入选项
14                     cnt++;
15             }
16    }
17    return cnt;
18 }

```

- ☐ A.  $O(N \log N)$
- ☐ B.  $O(N^{1/2})$
- ☐ C.  $O(N^{1/2} \log N)$
- ☐ D.  $O(N^{1/2}(\log N)^2)$

第 11 题 下面的欧氏筛法程序中，两个横线处应填入的分别是（ ）。

```

1  int primes[MAXP], num = 0;
2  bool isPrime[MAXN + 1] = {false};
3  void sieve() {
4      for (int n = 2; n <= MAXN; n++) {
5          if (!isPrime[n])
6              primes[num++] = n;
7          for (int i = 0; i < num && _____; i++) { // 在此处填入选项
8              isPrime[n * primes[i]] = true;
9              if (_____) // 在此处填入选项
10                 break;
11         }
12     }
13 }
```

☐ A.

```

1  n * primes[i] < MAXN
2  n % primes[i] == 0
```

☐ B.

```

1  n * primes[i] < MAXN
2  primes[i] > n
```

☐ C.

```

1  n * primes[i] <= MAXN
2  n % primes[i] == 0
```

☐ D.

```

1  n * primes[i] <= MAXN
2  primes[i] > n
```

第 12 题 下面Floyd算法中，横线处应该填入的是（ ）。

```

1  #include <iostream>
2  using namespace std;
3
4  #define N 21
5  #define INF 99999999
6  int map[N][N];
7  int main() {
8      int n, m, t1, t2, t3;
9      cin >> n >> m;
```

```

10     for (int i = 1; i <= n; i++) {
11         for (int j = 1; j <= n; j++) {
12             if (i == j)
13                 map[i][j] = 0;
14             else
15                 map[i][j] = INF;
16         }
17     }
18     for (int i = 1; i <= m; i++) {
19         cin >> t1 >> t2 >> t3;
20         map[t1][t2] = t3;
21     }
22     for (int k = 1; k <= n; k++)
23         for (int i = 1; i <= n; i++)
24             for (int j = 1; j <= n; j++)
25                 if (map[i][j] > map[i][k] + map[k][j])
26                     _____; // 在此处填入选项
27     for (int i = 1; i <= n; i++) {
28         for (int j = 1; j <= n; j++) {
29             cout.width(4);
30             cout << map[i][j];
31         }
32         cout << endl;
33     }
34 }

```

☐ A.

```
1 map[i][j] = map[i][k] + map[k][j]
```

☐ B.

```
1 map[i][k] = map[i][j] - map[k][j]
```

☐ C.

```
1 map[i][j] = map[i][k] - map[k][j]
```

☐ D.

```
1 map[k][j] = map[i][j] - map[i][k]
```

第 13 题 下面Floyd算法程序的时间复杂度为（ ）。

```

1  #include <iostream>
2  using namespace std;
3
4  #define N 21
5  #define INF 99999999
6  int map[N][N];
7  int main() {
8      int n, m, t1, t2, t3;
9      cin >> n >> m;
10     for (int i = 1; i <= n; i++) {

```

```

12         if (i == j)
13             map[i][j] = 0;
14         else
15             map[i][j] = INF;
16     }
17 }
18 for (int i = 1; i <= m; i++) {
19     cin >> t1 >> t2 >> t3;
20     map[t1][t2] = t3;
21 }
22 for (int k = 1; k <= n; k++)
23     for (int i = 1; i <= n; i++)
24         for (int j = 1; j <= n; j++)
25             if (map[i][j] > map[i][k] + map[k][j])
26                 _____; // 在此处填入选项
27 for (int i = 1; i <= n; i++) {
28     for (int j = 1; j <= n; j++) {
29         cout.width(4);
30         cout << map[i][j];
31     }
32     cout << endl;
33 }
34 }

```

- ☐ A.  $O(N)$
- ☐ B.  $O(N^2)$
- ☐ C.  $O(N^3)$
- ☐ D.  $O(N^2 \log N)$

第 14 题 下列程序实现了输出杨辉三角形，代码中横线部分应该填入的是（ ）。

```

1  #include <iostream>
2  using namespace std;
3  #define N 35
4  int a[N];
5  int main() {
6      int n;
7      cin >> n;
8      for (int i = 0; i < n; i++) {
9          a[i] = 1;
10         for (int j = i - 1; j > 0; j--)
11             _____; // 在此处填入选项
12         for (int j = 0; j <= i; j++)
13             cout << a[j] << " ";
14         cout << endl;
15     }
16     return 0;
17 }

```

- ☐ A.

1 | a[j] += a[j + 1]

☐ B.

```
1 | a[j] += a[j - 1]
```

☐ C.

```
1 | a[j - 1] += a[j]
```

☐ D.

```
1 | a[j + 1] += a[j]
```

第 15 题 下列程序实现了输出杨辉三角形，其时间复杂度为（ ）。

```
1 | #include <iostream>
2 | using namespace std;
3 | #define N 35
4 | int a[N];
5 | int main() {
6 |     int n;
7 |     cin >> n;
8 |     for (int i = 0; i < n; i++) {
9 |         a[i] = 1;
10 |        for (int j = i - 1; j > 0; j--)
11 |            _____; // 在此处填入选项
12 |        for (int j = 0; j <= i; j++)
13 |            cout << a[j] << " ";
14 |        cout << endl;
15 |    }
16 |    return 0;
17 | }
```

☐ A.  $O(n)$

☐ B.  $O(n \log n)$

☐ C.  $O(n^2)$

☐ D.  $O(n^3)$

## 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 表达式 '5' - 3.0 的结果为 2.0，类型为 double。

第 2 题 在 C++ 语言中，如果想要在一个函数内调用一个类的私有方法，可以在该类中将该函数声明为友元函数。

第 3 题 插入排序一般是稳定的。

第 4 题 5 个相同的红球和 4 个相同的蓝球排成一排，要求蓝球不能相邻，则一共有 15 种排列方案。

第 5 题 使用 math.h 或 cmath 头文件中的函数，表达式 pow(2, 5) 的结果类型为 int、值为 32。

**第 6 题** C++是一种面向对象编程语言，C则不是。多态是面向对象三大特性之一，虚函数是动态多态的代表特性。因此，使用C语言无法实现虚函数。

**第 7 题** 在 $N$ 个节点的平衡二叉树中查找指定元素的最差时间复杂度为 $O(N)$ 。

**第 8 题** 定义 `int` 类型的变量 `a` 和 `b`，求二次函数 $y = x^2 + ax + b$ 取最小值时 $x$ 的值，可以通过表达式 `-a / 2.0` 求得。

**第 9 题** 判断无向图中是否有环，可以通过广度优先搜索实现。

**第 10 题** 从32名学生中选出4人分别担任班长、副班长、学习委员和组织委员，共有 $C(32, 4)$ 种不同的选法。