

2025 年转专业试题（回忆版）

57-程序设计与算法语言

2025.4

回忆人：24 级 睦恂@w

24 级 jllll(讨厌 b3 版)

一、读程序写结果（6*5%=30%）

1.

```
#include <iostream>
using namespace std;

int v = 5;

int main() {
    int v = 65;
    for (int v = 65; v <= 66; v++) {
        switch (v) {
            case 'A': cout << 'A' << endl; break;
            case 'B': cout << 'B' << endl;
            default: cout << "::v << endl;
        }
    }
    cout << v << endl;
    return 0;
}
```

2.

```
#include <iostream>
using namespace std;
int func(int* b)
{
    return b[3];
}

int a[][2] = { {3}, {2,1} };

int main()
{
    cout << func((int*)a) << endl;
    return 0;
}
```

3.

```
#include <iostream>
using namespace std;

int func(int a, int& r) {
    int c = 3;
    static int b = 0;
    b += 1;
    c += 1;
    r += a + b + c;
    return r;
}

int main() {
    int a = 2, y = 0;
    for (int m = 1; m <= 4; m++) {
        cout << func(a, y) << endl;
    }
    cout << endl;
    cout << y << endl;
    return 0;
}
```

4.

```
#include <iostream>
using namespace std;

class Test
{
private:
    static int count;
public:
    Test() { count++; }
    ~Test() { count--; }

    void display()
    {
        cout << count << endl;
    }
};

int Test::count = 0;
Test m1;
void display_count(Test* m_ptr){
    Test m3;
```

```

    Test* s_ptr = new Test[5];
    m3.display();
    delete[] s_ptr;
    s_ptr = nullptr;
    delete m_ptr;
    m_ptr = nullptr;
}

```

```

int main()
{
    m1.display();
    Test m2;
    Test* m_ptr = new Test();
    m_ptr->display();
    display_count(m_ptr);
    m2.display();
    Test m3(m2);
    m3.display();
    return 0;
}

```

5.

```

#include<iostream>
using namespace std;
int fun(int n)
{
    if (n == 0) { return 1; }
    else {
        int res = n * fun(n - 1);
        cout << res << endl;
        return res;
    }
}

int main() {
    cout << fun(3) << endl;
    return 0;
}

```

6.

```
#include <iostream>
using namespace std;

void func(int arr[][2]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            if (i == 0 && j == 0)
                arr[i][j] = arr[i][j] * arr[1][1];
            else if (i == 1 && j == 1)
                arr[i][j] = arr[i][j] * arr[0][0];
            else
                arr[i][j] = -arr[i][j];
        }
    }
}

int main() {
    int arr[2][2] = { {1,2}, { 3,4 } };
    func(arr);
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            cout << arr[i][j] << ' ';
        }
        cout << endl;
    }
    cout << endl;
}
```

二、补全程序 (2*10%=20%)

1.

```
#include <iostream>
using namespace std;

bool isMatch(const char* a, const char* sub)
{
    if (*sub == '\0')
        return true;
    if (*a == '\0')
        return false;
    if (*a == *sub)//空(1)if(____)
        return isMatch(a + 1, sub + 1);//空(2)
}
```

```

        return false;
    }

int find_substr(const char* a, const char* sub, int index = 0)
{
    if (*a == '\0')//空(3)if(____)
        return -1;
    if (isMatch(a, sub))//空(4)if(____)
        return index;
    return find_substr(a + 1, sub, index + 1);
}

int main()
{
    const char* a = "hello world";
    const char* sub = "world";
    int position = find_substr(a, sub);//空(5)int position =____;
    if (position != -1)
    {
        cout << position;
    }
    else
        cout << "找不到" << endl;
    return 0;
}

```

2.

```

#include <iostream>
#include <cstring>
using namespace std;

class Book {
private:
    string title;
    double price;
public:
    Book(string t, double p) { //空(1) _____(string t, double p){
        title = t; //空(2) _____=t;
        price = p; //空(3) _____=p;
    }
    ~Book() { //空(4) _____(){
        cout << "the object is destroying!\n";
    }
    void Output() {

```

```

        cout << "The title of the book is:" << title << endl;
        cout << "The price of the book is:" << price << endl;
    }
};

int main() {
    Book b1("C Plus Premier", 39.9);
    b1.Output();
    return 0;
}

```

三、编写程序 (2*20%=40%)

1.考的向量相似度

具体实现：对于两个手机号字符串（函数 1），按照字符串中数字出现频率来构建向量，并计算两个向量的相似度，并写出 main 函数进行测试。

用频率构建两个手机号字符串的向量，计算两个向量的向量相似度（函数 2），用 main 函数测试。

向量相似度可以用两个向量夹角的余弦表示，可以写成向量内积与向量长度之积的比值，计算公式：

$$\text{向量相似度} = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}| \cdot |\mathbf{B}|}$$

向量构建(8%)计算相似度(7%)主函数测试(5%)

2.字符串数组

给定一个字符串数组，const char* phones[5]{"13899848445", "13899948564", "13813915644", "13813614565", "13813888888"}，写一个排序函数，将字符串按照字典顺序排序，并写出输出函数，写 main 函数进行测试（可以使用 cstring 的库函数）

排序函数(8%)输出函数(6%)主函数测试(6%)

四、简答题 (10%)

C++中有值传递、指针传递和引用传递，请简要说说三者的不同之处并举例说明。