

计算机科学基础 I——homework6

说明

- 代码阅读题请勿借助计算机直接运行程序，应通过自行阅读与分析得出结果；
- 若需要换行，输出写成多行；若代码语法有问题，请在输出结果处写 error；若没有输出写 /；若死循环写死循环；
- 程序编写及调试请使用 Visual C++；
- 提交作业时将工程文件与包含代码阅读题目答案的 pdf 文件打包压缩发送到邮箱: seuiccpp@163.com, 压缩包命名为“学号-姓名-hw6”；
- 截止时间: 2025 年 12 月 12 日

代码阅读

1. 阅读下面程序，写出程序的输出结果。

```
1 void f(int x) {
2     x = x + 10;
3     cout << "x = " << x << endl;
4 }
5
6 int main() {
7     int x = 5;
8     cout << "x = " << x << endl;
9     f(x);
10    cout << "x = " << x << endl;
11    return 0;
12 }
```

2. 阅读下面程序，写出程序的输出结果。

```

1 void change(int a, int b) {
2     int temp = a;
3     a = b;
4     b = temp;
5     cout << "inside: " << a << " " << b << endl;
6 }
7
8 int main() {
9     int x = 3, y = 7;
10    cout << "before: " << x << " " << y << endl;
11    change(x, y);
12    cout << "after: " << x << " " << y << endl;
13    return 0;
14 }
```

3. 阅读下面程序，写出程序的输出结果。

```

1 int main() {
2     int a = 1;
3     cout << a << endl;
4     {
5         int a = 2;
6         cout << a << endl;
7         {
8             int a = 3;
9             cout << a << endl;
10            }
11            cout << a << endl;
12        }
13        cout << a << endl;
14        return 0;
15 }
```

4. 阅读下面程序，写出程序的输出结果。

```

1 void count() {
2     int x = 0;
3     x++;
4     cout << x << " ";
5 }
```

```

7 int main() {
8     for (int i = 0; i < 3; i++) {
9         count();
10    }
11    return 0;
12 }
```

5. 阅读下面程序，写出程序的输出结果。

```

1 void f(int x) {
2     cout << "f: " << x << endl;
3     x = x + 5;
4     cout << "f: " << x << endl;
5 }
6
7 void g(int x) {
8     cout << "g: " << x << endl;
9     f(x);
10    cout << "g: " << x << endl;
11 }
12
13 int main() {
14     int n = 10;
15     cout << "main: " << n << endl;
16     g(n);
17     cout << "main: " << n << endl;
18     return 0;
19 }
```

6. 阅读下面程序，写出程序的输出结果。

```

1 int main() {
2     int x = 5;
3     if (x > 0) {
4         int y = x + 1;
5         cout << x << " " << y << endl;
6     }
7     cout << x << endl;
8     y = 10;
9     cout << y << endl;
10    return 0;
11 }
```

7. 阅读下面程序，写出程序的输出结果。

```
1 void print(int n) {
2     if (n == 0) {
3         cout << "end" << endl;
4         return;
5     }
6     cout << n << " ";
7     print(n - 1);
8     cout << n << " ";
9 }
10
11 int main() {
12     print(3);
13     return 0;
14 }
```

8. 阅读下面程序，写出程序的输出结果。

```
1 int f(int n) {
2     if (n <= 1) {
3         cout << "f(" << n << ")" = " << n << endl;
4         return n;
5     }
6     int t = f(n - 1);
7     int r = t + n;
8     cout << "f(" << n << ")" = " << r << endl;
9     return r;
10 }
11
12 int main() {
13     int result = f(3);
14     cout << "result=" << result << endl;
15     return 0;
16 }
```

9. 阅读下面程序，写出程序的输出结果。

```
1 int x = 1;
2
3 void f(int x) {
4     cout << x << endl;
5 }
6
7 int main() {
8     cout << x << endl;
9     f(2);
10    cout << x << endl;
11    return 0;
12 }
```

10. 阅读下面程序，写出程序的输出结果。

```
1 int sum(int n) {
2     if (n == 0) {
3         return 0;
4     }
5     return n + sum(n - 1);
6 }
7
8 int main() {
9     cout << sum(3) << endl;
10    cout << sum(0) << endl;
11    return 0;
12 }
```

11. 阅读下面程序，写出程序的输出结果。

```
1 void down(int n) {
2     if (n <= 0) {
3         return;
4     }
5     cout << n << " ";
6     down(n - 2);
7 }
8
9 int main() {
10    down(5);
11    return 0;
12 }
```

12 }

12. 阅读下面程序，写出程序的输出结果。

```
1 int main() {  
2     int a = 1;  
3     int a = 2;  
4     cout << a << endl;  
5     return 0;  
6 }
```

13. 阅读下面程序，写出程序的输出结果。

```
1 int main() {  
2     int x = 0;  
3     for (int i = 0; i < 3; i++) {  
4         int x = i;  
5         cout << x << " "  
6     }  
7     cout << x << endl;  
8     return 0;  
9 }
```

14. 阅读下面程序，写出程序的输出结果。

```
1 int f(int x) {  
2     x = x + 1;  
3     return x;  
4 }  
5  
6 int g(int x) {  
7     x = f(x);  
8     x = f(x);  
9     return x;  
10 }  
11  
12 int main() {  
13     int a = 1;  
14     int b = g(a);  
15     cout << a << " "  
16     << b << endl;  
17     return 0;  
18 }
```

15. 阅读下面程序，写出程序的输出结果。

```
1 int add_one(int x);  
2  
3 int main() {  
4     int x = 10;  
5     {  
6         int x = 20;  
7         cout << add_one(x) << endl;  
8     }  
9     cout << add_one(x) << endl;  
10    return 0;  
11}  
12  
13 int add_one(int x) {  
14     return x + 1;  
15}
```

16. 阅读下面程序，写出程序的输出结果。

```
1 int mystery(int n) {  
2     if (n == 1) {  
3         cout << "m(" << n << ") ";  
4         return 1;  
5     }  
6     int t = mystery(n - 1);  
7     cout << "m(" << n << ") ";  
8     return t * n;  
9 }  
10  
11 int main() {  
12     int r = mystery(3);  
13     cout << endl << r << endl;  
14     return 0;  
15 }
```

代码填空

1. 下列程序中，函数 `max` 的声明和定义不完整。请在代码中的空白处填入合适的内容，使程序能够正确编译并输出两个整数中的较大者。

```
1 // (1) 在此处补全
2
3 int main() {
4     int x, y;
5     cin >> x >> y;
6     cout << "max = " << max(x, y) << endl;
7     return 0;
8 }
9
10 // (2) 在此处补全
11 ----- max(-----) {
12     // (3) 在此处补全
13     if (a > b) return a;
14     else return b;
15 }
```

2. 下列程序试图通过自定义函数 `square` 计算一个浮点数的平方。请在空白处填入合适的内容。

```
1 // (1) 在此处补全  
2  
3 int main() {  
4     double n;  
5     cin >> n;  
6     // (2) 在此处补全  
7     cout << square(-----) << endl;  
8     return 0;  
9 }  
10  
11 // (3) 在此处补全  
12 ----- square(-----) {  
13     // (4) 在此处补全  
14     return -----;  
15 }
```

3. 下列程序试图使用递归函数 `fibonacci` 计算斐波那契数列的第 n 项。请在空白处填入合适的内容，使程序正确计算并输出结果（假设输入 $n \geq 0$ ）。

```
1 int fibonacci(int n) {  
2     // (1) 在此处补全  
3     if (-----) return -----;  
4     // (2) 在此处补全  
5     return ----- + -----;  
6 }  
7  
8 int main() {  
9     int n;  
10    cin >> n;  
11    cout << fibonacci(n) << endl;  
12    return 0;  
13 }
```

4. 下列程序试图使用递归函数 `sumToN` 计算 $1 + 2 + \dots + n$ 。请在空白处填入合适的内容。

```
1 int sumToN(int n) {  
2     // (1) 在此处补全  
3     if (-----) return -----;
```

```

4 // (2) 在此处补全
5     return ----- + sumToN(-----);
6 }
7
8 int main() {
9     int n;
10    cin >> n;
11    cout << sumToN(n) << endl;
12    return 0;
13 }
```

5. 下列程序试图使用递归函数 `printDigits` 从高位到低位输出一个非负整数 `n` 的每一位数字，中间用空格分隔。例如输入 1234，输出为 1 2 3 4。请在空白处填入合适的内容。

```

1 void printDigits(int n) {
2     // (1) 在此处补全
3     if (-----) {
4         cout << n;
5     }
6     else {
7         // (2) 在此处补全
8         printDigits(-----);
9         // (3) 在此处补全
10        cout << " " << -----;
11    }
12 }
13
14 int main() {
15     int n;
16     cin >> n;
17     printDigits(n);
18     cout << endl;
19     return 0;
20 }
```

代码编写

题目 1：综合数位规则判断 `isStrongNumber`

请编写一个函数 `bool isStrongNumber(int n);`。若给定的正整数 n 满足下列所有规则，则返回 `true`，否则返回 `false`。

- **基本合法性**若 $n \leq 0$ ，返回 `false`。
- **条件 A：数位和是质数**设 S 为 n 的数位和，例如

$$472 \rightarrow 4 + 7 + 2 = 13.$$

若 S 为质数，则通过；否则失败。

- **条件 B：非零数位积的特殊整除关系**设 P 为 n 的所有非零数字的乘积，例如

$$4025 \rightarrow 4 \cdot 2 \cdot 5 = 40.$$

要求：

$$P \bmod 3 = 0, \quad P \bmod 9 \neq 0.$$

- **条件 C：相邻数字差值限制**对 n 的任意一对相邻数字 d_{left} (左)、 d_{right} (右) 要求：

$$|d_{\text{right}} - d_{\text{left}}| \leq 3.$$

若存在一对差值大于 3，则失败。

- **条件 D：位数为偶数且最高位为奇数**设 cnt 为 n 的位数， $firstDigit$ 为最高位数字。要求：

$$cnt \text{ 为偶数}, \quad firstDigit \text{ 为奇数}.$$

若上述所有条件全部满足，则函数返回 `true`，否则返回 `false`。

题目 2: Tribonacci 数列的递归求解

定义一个 Tribonacci 数列，其递推关系如下：

$$T(0) = 0, \quad T(1) = 1, \quad T(2) = 1,$$

$$T(n) = T(n - 1) + T(n - 2) + T(n - 3), \quad n \geq 3.$$

请分别编写一个基于递归的函数 `long long trib_rec(int n)` 来计算 $T(n)$ 和基于循环的函数 `long long trib_iter(int n)` 来计算 $T(n)$ ，并对比随着 n 的增大，两个函数的运行效率有什么变化，分析其原因。