

08 - 多重循环

C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 5 月 17 日

目录

1 复习回顾

2 多重循环

3 多重循环的应用

4 break 和 continue 语句

5 总结

例题回顾 - 加减数列求和

编程题

- 编写程序，输入一个整数 n ($1 \leq n \leq 10^6$)，求 $1 - 2 + 3 - 4 + 5 \dots$ ，一直到 n 的和。
- 样例输入
7
- 样例输出
4
- 解释说明
 $1 - 2 + 3 - 4 + 5 - 6 + 7 = 4$

例题回顾 - 加减数列求和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, sum = 0;
7     cin >> n;
8     for (int i = 1; i <= n; i++) {
9         if (i % 2 == 0) sum += i;
10        else sum -= i;
11    }
12    cout << sum << endl;
13    return 0;
14 }
```

既然循环体中可以写顺序结构和分支结构，
那么是否可以写循环结构呢？

目录

1 复习回顾

2 多重循环

3 多重循环的应用

4 break 和 continue 语句

5 总结

多重循环

- 一重循环

多重循环

- 一重循环
- 二重循环

多重循环

- 一重循环
- 二重循环
 - 在一重循环的循环体中再写一个循环

多重循环

- 一重循环
- 二重循环
 - 在一重循环的循环体中再写一个循环
- 三重循环

多重循环

- 一重循环
- 二重循环
 - 在一重循环的循环体中再写一个循环
- 三重循环
- ...

例 8.1.1: 输出一行 m 个星号

编程题

- 编写程序, 输入一个整数 m ($1 \leq m \leq 1000$), 输出一行 m 个星号。
- 样例输入
5
- 样例输出

例 8.1.1: 输出一行 m 个星号

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int m;
7     cin >> m;
8     for (int i = 1; i <= m; i++) {
9         cout << "*";
10    }
11    cout << endl;
12    return 0;
13 }
```

例 8.1.2: 输出 n 行 m 个星号

编程题

- 编写程序，输入两个整数 n, m ($1 \leq n, m \leq 1000$), 输出 n 行 m 个星号。
- 样例输入
3 5
- 样例输出

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入
2 3
- 变量
i = 1

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 1

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 1

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 1

- 输出

*

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入
2 3
- 变量
i = 1
j = 2
- 输出
*

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 2

- 输出

*

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 2

- 输出

* *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 3

- 输出

* *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 3

- 输出

* *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 3

- 输出

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 1

j = 4

- 输出

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入
2 3
- 变量
i = 1
j <= 3 不成立
- 输出
* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入
2 3
- 变量
i = 1
- 输出
* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入
2 3
- 变量
 $i = 2$
- 输出
* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入
2 3
- 变量
i = 2
- 输出
* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 1

- 输出

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 1

- 输出

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 1

- 输出

* * *

*

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 2

- 输出

* * *

*

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 2

- 输出

* * *

*

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 2

- 输出

* * *

* *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 3

- 输出

* * *

* *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 3

- 输出

* * *

* *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 3

- 输出

* * *

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

j = 4

- 输出

* * *

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入
2 3
- 变量
i = 2
j <= 3 不成立
- 输出
* * *
* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 2

- 输出

* * *

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

i = 3

- 输出

* * *

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

$i \leq 2$ 不成立

- 输出

* * *

* * *

示例：双重循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "*";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

- 输入

2 3

- 变量

- 输出

* * *

* * *

目录

1 复习回顾

2 多重循环

3 多重循环的应用

4 break 和 continue 语句

5 总结

例 8.2: 输出 n 行从 1 到 m 的整数

编程题

- 编写程序, 输入两个整数 n, m ($1 \leq n, m \leq 1000$), 输出 n 行从 1 到 m 的整数。
- 样例输入
2 4
- 样例输出
1 2 3 4
1 2 3 4

例 8.2：输出 n 行从 1 到 m 的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << j << " ";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

例 8.3：输出二重循环的循环变量

编程题

- 在【例 8.2】的基础上，输出二重循环的循环变量 (i, j) 的值，每次输出以空格间隔。
- 样例输入
2 4
- 样例输出
(1,1) (1,2) (1,3) (1,4)
(2,1) (2,2) (2,3) (2,4)

例 8.3：输出二重循环的循环变量

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             cout << "(" << i << " " << j << ") ";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1) (1,2) (1,3) (1,4)

(2,1) (2,2) (2,3) (2,4)

(3,1) (1,2) (1,3) (1,4)

- i 表示所在的行号
- j 表示所在的列号

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1) (1,2) (1,3) (1,4) $i = 1$

(2,1) (2,2) (2,3) (2,4)

(3,1) (1,2) (1,3) (1,4)

- i 表示所在的行号
- j 表示所在的列号

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1) (1,2) (1,3) (1,4) $i = 2$

(2,1) (2,2) (2,3) (2,4)

(3,1) (3,2) (3,3) (3,4)

- i 表示所在的行号
- j 表示所在的列号

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1) (1,2) (1,3) (1,4) $i = 3$

(2,1) (2,2) (2,3) (2,4)

(3,1) (1,2) (1,3) (1,4)

- i 表示所在的行号
- j 表示所在的列号

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(1,2)	(1,3)	(1,4)

$j = 1$

- i 表示所在的行号
- j 表示所在的列号

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(1,2)	(1,3)	(1,4)

$j = 2$

- i 表示所在的行号
- j 表示所在的列号

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)

$j = 3$

- i 表示所在的行号
- j 表示所在的列号

例 8.4：输出二重循环的循环变量

- 当输入 n 为 3, m 为 4 时, 输出如图所示

3 4

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(1,2)	(1,3)	(1,4)

$j = 4$

- i 表示所在的行号
- j 表示所在的列号

例 8.5：输出数字直角三角形

编程题

- 编写程序，输入一个整数 n ($1 \leq n \leq 10^3$)，输出一个等腰直角三角形，腰长为 n ，每行由整数 $1, 2, 3 \dots$ 组成，整数之间用空格间隔。
- 样例输入
3
- 样例输出
1
1 2
1 2 3

例 8.5：输出数字直角三角形

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= i; j++) {
10             cout << j << " ";
11         }
12         cout << endl;
13     }
14     return 0;
15 }
```

例 8.6：累加数列求和

编程题

- 已知 $A_i = 1 + 2 + 3 + \dots + i$, $S_n = A_1 + A_2 + A_3 + \dots + A_n$ 。编写程序，输入一个整数 n ($1 \leq n \leq 10^4$)，计算 S_n 的结果并输出。
- 样例输入
3
- 样例输出
10
- 样例解释
 $A_1 = 1$
 $A_2 = 1 + 2$
 $A_3 = 1 + 2 + 3$
 $S_3 = A_1 + A_2 + A_3$

例 8.6：累加数列求和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, Ai;
7     long long Sn = 0;
8     cin >> n;
9     for (int i = 1; i <= n; i++) {
10         // 用于累加 1~i 的累加器 Ai, 需要初始化为 0
11         Ai = 0;
12         for (int j = 1; j <= i; j++) {
13             Ai += j;
14         }
15         Sn += Ai; // 将求得的 Ai 累加到 Sn 中
16     }
17     return 0;
18 }
```

例 8.6：累加数列求和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, Ai;
7     long long Sn = 0;
8     cin >> n;
9     for (int i = 1; i <= n; i++) {
10         // 用于累加 1~i 的累加器 Ai, 需要初始化为 0
11         Ai = 0;
12         for (int j = 1; j <= i; j++) {
13             Ai += j;
14         }
15         Sn += Ai; // 将求得的 Ai 累加到 Sn 中
16     }
17     return 0;
18 }
```


例 8.6：累加数列求和 - 代码优化

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, Ai = 0;
7     long long Sn = 0;
8     cin >> n;
9     for (int i = 1; i <= n; i++) {
10         Ai += i; // 当前的 Ai 为 A(i-1) + i
11         Sn += Ai; // 将求得的 Ai 累加到 Sn 中
12     }
13     return 0;
14 }
```

例 8.7: 空心矩形

编程题

- 编写程序，输入两个整数 n, m ($1 \leq n, m \leq 10^3$), 输出一个 n 行 m 列的空心矩形，矩形由星号形成。

- 样例输入

4 7

- 样例输出

* *

* *

例 8.7: 空心矩形

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             if (i == 1 || i == n || j == 1 || j == m) {
11                 cout << "*";
12             } else {
13                 cout << " ";
14             }
15         }
16         cout << endl;
17     }
18     return 0;
19 }
```

例 8.7: 空心矩形

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= m; j++) {
10             if (i == 1 || i == n || j == 1 || j == m) {
11                 cout << "*";
12             } else {
13                 cout << " ";
14             }
15         }
16         cout << endl;
17     }
18     return 0;
19 }
```

例 8.8：等腰三角形

编程题

- 编写程序，输入一个整数 n ($1 \leq n \leq 20$), 输出高为 n 的由 * 号组成的等腰三角形。

- 样例输入

3

- 样例输出

*

例 8.8：等腰三角形

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n;
5     cin >> n;
6     // kg 表示每行要输出的空格数量
7     // xh 表示每行要输出的星号数量
8     int kg = n - 1, xh = 1;
9     for (int i = 1; i <= n; i++) {
10         for (int j = 1; j <= kg; j++) { // 输出 kg 空格
11             cout << " ";
12         }
13         for (int j = 1; j <= xh; j++) { // 输出 xh 星号
14             cout << "*";
15         }
16         cout << endl;
17         kg--, xh += 2; // 下一行空格数 - 1, 星号数 + 2
18     }
19     return 0;
20 }
```

例 8.9：求一个整数中含有多少个 1

编程题

- 编写程序，输入一个整数 n ($1 \leq n \leq 10^9$)，统计 n 的数位上有多少个 1，输出 n 和其含有 1 的个数，用空格间隔。
- 样例输入
1351
- 样例输出
1351 2

例 8.9：求一个整数中含有多少个 1

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     do {
7         int r = n % 10;
8         n /= 10;
9         if (r == 1) cnt++;
10    } while (n);
11    cout << n << " " << cnt << endl;
12    return 0;
13 }
```


例 8.9：求一个整数中含有多少个 1

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     do {
7         int r = n % 10;
8         n /= 10;
9         if (r == 1) cnt++;
10    } while (n);
11    cout << n << " " << cnt << endl;
12    return 0;
13 }
```

- 运行结果正确吗？

例 8.9：求一个整数中含有多少个 1

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     do {
7         int r = n % 10;
8         n /= 10;
9         if (r == 1) cnt++;
10    } while (n);
11    cout << n << " " << cnt << endl;
12    return 0;
13 }
```

- 运行结果正确吗？
- 不正确，
数位拆分后 n 的值
会变为 0。

例 8.9：求一个整数中含有多少个 1

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     int tmp = n;
7     do {
8         int r = tmp % 10;
9         tmp /= 10;
10        if (r == 1) cnt++;
11    } while (tmp);
12    cout << n << " " << cnt << endl;
13    return 0;
14 }
```

例 8.9：求一个整数中含有多少个 1

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     int tmp = n;
7     do {
8         int r = tmp % 10;
9         tmp /= 10;
10        if (r == 1) cnt++;
11    } while (tmp);
12    cout << n << " " << cnt << endl;
13    return 0;
14 }
```

- 由于后续需要输出 n，因此不能直接用 n 进行数位拆分

例 8.10: 数 1 的个数

编程题

- 编写程序, 输入一个整数 n ($1 \leq n \leq 10^6$), 求从 1 到 n 的所有整数中, 数位上出现数字“1”的个数。
- 样例输入
12
- 样例输出
5
- 样例解释
当 $n = 12$ 时, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 中数位上出现了 5 个数字“1”。

例 8.10: 数 1 的个数

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     for (int i = 1; i <= n; i++) {
7         do {
8             int r = i % 10;
9             i /= 10;
10            if (r == 1) cnt++;
11        } while (i);
12    }
13    cout << cnt << endl;
14    return 0;
15 }
```

例 8.10: 数 1 的个数

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     for (int i = 1; i <= n; i++) {
7         do {
8             int r = i % 10;
9             i /= 10;
10            if (r == 1) cnt++;
11        } while (i);
12    }
13    cout << cnt << endl;
14    return 0;
15 }
```

- 运行结果正确吗?

例 8.10: 数 1 的个数

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     for (int i = 1; i <= n; i++) {
7         do {
8             int r = i % 10;
9             i /= 10;
10            if (r == 1) cnt++;
11        } while (i);
12    }
13    cout << cnt << endl;
14    return 0;
15 }
```

- 运行结果正确吗?
- 不正确, 会死循环。
因为数位拆分后 i 会变化!

例 8.10: 数 1 的个数

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     for (int i = 1; i <= n; i++) {
7         int tmp = i;
8         do {
9             int r = tmp % 10;
10            tmp /= 10;
11            if (r == 1) cnt++;
12        } while (tmp);
13    }
14    cout << cnt << endl;
15    return 0;
16 }
```

例 8.10: 数 1 的个数

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n, cnt = 0;
5     cin >> n;
6     for (int i = 1; i <= n; i++) {
7         int tmp = i;
8         do {
9             int r = tmp % 10;
10            tmp /= 10;
11            if (r == 1) cnt++;
12        } while (tmp);
13    }
14    cout << cnt << endl;
15    return 0;
16 }
```

- 为了避免程序进入死循环，使用临时变量 tmp 进行数位拆分

目录


- 1 复习回顾
- 2 多重循环
- 3 多重循环的应用
- 4 break 和 continue 语句**
- 5 总结

break 在多重循环中的应用

```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             break;  
5         }  
6     }  
7     ...  
8 }
```

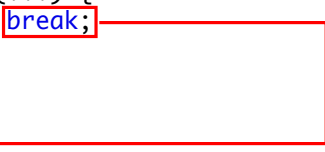
break 在多重循环中的应用

```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             break;  
5         }  
6     }  
7     ...  
8 }
```



break 在多重循环中的应用

```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             break;  
5         }  
6     }  
7     ...  
8 }
```

A red rectangular box highlights the `break;` statement on line 4. A red line extends from the box to the right, then turns downwards and then left, ending with an arrow pointing to the closing brace of the outer loop on line 8. This illustrates that the `break` statement exits the innermost loop.

```
1 for (int i = 1; i <= n; i++) {  
2     if (...) {  
3         break;  
4     }  
5     for (int j = 1; j <= m; j++) {  
6         ...  
7     }  
8     ...  
9 }
```

break 在多重循环中的应用

```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             break;  
5         }  
6     }  
7     ...  
8 }
```

A red box highlights the `break;` statement on line 4. A red line starts from the box, goes right, then down, then left, ending with an arrow pointing to the closing brace of the inner loop on line 6. This illustrates that `break` exits the innermost loop.

```
1 for (int i = 1; i <= n; i++) {  
2     if (...) {  
3         break;  
4     }  
5     for (int j = 1; j <= m; j++) {  
6         ...  
7     }  
8     ...  
9 }
```

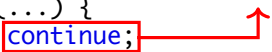
A red box highlights the `break;` statement on line 3. A red line starts from the box, goes right, then down, then left, ending with an arrow pointing to the closing brace of the outer loop on line 9. This illustrates that `break` exits the loop it is currently in, which in this case is the outer loop.

continue 在多重循环中的应用

```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             continue;  
5         }  
6     }  
7     ...  
8 }
```

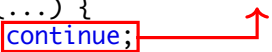

continue 在多重循环中的应用

```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             continue;  
5         }  
6     }  
7     ...  
8 }
```



continue 在多重循环中的应用

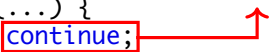
```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             continue;  
5         }  
6     }  
7     ...  
8 }
```




```
1 for (int i = 1; i <= n; i++) {  
2     if (...) {  
3         continue;  
4     }  
5     for (int j = 1; j <= m; j++) {  
6         ...  
7     }  
8     ...  
9 }
```

continue 在多重循环中的应用

```
1 for (int i = 1; i <= n; i++) {  
2     for (int j = 1; j <= m; j++) {  
3         if (...) {  
4             continue;  
5         }  
6     }  
7     ...  
8 }
```



```
1 for (int i = 1; i <= n; i++) {  
2     if (...) {  
3         continue;  
4     }  
5     for (int j = 1; j <= m; j++) {  
6         ...  
7     }  
8     ...  
9 }
```



目录

- 1 复习回顾
- 2 多重循环
- 3 多重循环的应用
- 4 break 和 continue 语句
- 5 总结

- 多重循环的概念

总结

- 多重循环的概念
- 多重循环的应用

- 多重循环的概念
- 多重循环的应用
 - 输出图形问题

- 多重循环的概念
- 多重循环的应用
 - 输出图形问题
 - 数列计算问题

- 多重循环的概念
- 多重循环的应用
 - 输出图形问题
 - 数列计算问题
 - 数位拆分问题