

# 07 - 循环与分支综合

## C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 7 月 20 日

- 1 复习回顾
- 2 for 循环嵌套分支
- 3 while/do..while 循环嵌套分支
- 4 break 和 continue 语句
- 5 总结

# 分支结构

- 根据不同情况，执行不同的语句

# 分支结构

- 根据不同情况，执行不同的语句
  - 单分支

```
1 if (条件语句) {  
2     语句块;  
3 }
```

# 分支结构

- 根据不同情况，执行不同的语句
  - 单分支
  - 双分支

```
1 if (条件语句) {  
2     语句块 1;  
3 } else {  
4     语句块 2;  
5 }
```

# 分支结构

- 根据不同情况，执行不同的语句
  - 单分支
  - 双分支
  - 多分支

```
1 if (条件语句 1) {  
2     语句块 1;  
3 } else if (条件语句 2) {  
4     语句块 2;  
5 } else if (条件语句 ...) {  
6     语句块 ...;  
7 } else {  
8     最后的语句块;  
9 }
```

# 分支结构

- 根据不同情况，执行不同的语句
  - 单分支
  - 双分支
  - 多分支
  - 分支嵌套

```
1 if (条件语句 1) {  
2     if (条件语句 2) {  
3         语句块 1;  
4     } else {  
5         语句块 2;  
6     }  
7 } else {  
8     if (条件语句 3) {  
9         语句块 3;  
10    } else {  
11        语句块 4;  
12    }  
13 }
```

# 问题回顾：闰年判断

## 编程题

- 闰年分为世纪闰年和普通闰年。普通闰年的年份是 4 的倍数，且不是 100 的倍数；世纪闰年的年份是 400 的倍数。  
输入一个整数  $year$  ( $1000 \leq year \leq 3000$ )，表示一个年份，判断该年份是否为闰年，输出对应的判断结果。
- 样例输入  
2010
- 样例输出  
2010 is not a leap year



# 问题回顾：闰年判断

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int y;
7     cin >> y;
8     // 闰年的判断条件:
9     // 年份是 4 的倍数且不是 100 的倍数 或 是 400 的倍数
10    if ((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) {
11        cout << y << " is a leap year" << endl;
12    } else {
13        cout << y << " is not a leap year" << endl;
14    }
15
16    return 0;
17 }
```

# 循环结构

- while 循环：先判断再执行

```
1 while (条件语句) {  
2     // 循环体  
3     每次执行的语句块;  
4 }
```

# 循环结构

- while 循环：先判断再执行
- do...while 循环：先执行再判断，至少执行一次

```
1 do {  
2     // 循环体  
3     每次执行的语句块;  
4 } while (条件语句);
```

# 循环结构

- while 循环：先判断再执行
- do...while 循环：先执行再判断，至少执行一次
- for 循环：执行初始动作后，先判断再执行

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2     // 每次循环执行的语句块  
3     循环体；  
4 }
```

# 问题回顾：输出从 1 到 $n$ 的整数之和

## 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出从 1 到  $n$  之间的整数之和。
- 样例输入  
5
- 样例输出  
15
- 样例说明  
 $1 + 2 + 3 + 4 + 5 = 15$

# 问题回顾：输出从 1 到 n 的整数之和

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

# 目录

1 复习回顾

**2 for 循环嵌套分支**

3 while/do..while 循环嵌套分支

4 break 和 continue 语句

5 总结

## 例 7.1：加减数列求和

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 - 2 + 3 - 4 + 5 - \dots$ , 一直到  $n$  的和。
- 样例输入  
7
- 样例输出  
4
- 样例说明  
 $1 - 2 + 3 - 4 + 5 - 6 + 7 = 4$



## 例 7.1：加減数列求和

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

## 例 7.2: 优秀学生

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 表示有  $n$  个学生。接下来输入每个学生的语文成绩  $x$  ( $1 \leq x \leq 100$ ) 和数学成绩  $y$  ( $1 \leq y \leq 100$ )。

对于每个学生, 如果其两科成绩均不低于 90 分, 则输出 Good Job, 否则输出 Fighting。

- 样例输入

```
2
98 92
80 95
```

- 样例输出

```
Good Job
Fighting
```

## 例 7.2: 优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

## 例 7.2：优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

- 允许边输入边输出

## 例 7.2：优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

- 允许边输入边输出
- 运行窗口

## 例 7.2: 优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

- 允许边输入边输出
- 运行窗口  
2

## 例 7.2: 优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

- 允许边输入边输出

- 运行窗口

```
2
98 92
```

## 例 7.2: 优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

- 允许边输入边输出

- 运行窗口

2

98 92

Good Job



## 例 7.2: 优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

- 允许边输入边输出

- 运行窗口

```
2
98 92
Good Job
80 95
```

## 例 7.2: 优秀学生

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int x, y;
9     for (int i = 1; i <= n; i++) {
10         cin >> x >> y;
11         if (x >= 90 && y >= 90) {
12             cout << "Good Job" << endl;
13         } else {
14             cout << "Fighting" << endl;
15         }
16     }
17
18     return 0;
19 }
```

- 允许边输入边输出
- 运行窗口  
2  
98 92  
Good Job  
80 95  
Fighting

## 例 7.3：输出 $n$ 的所有因子

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ )，输出  $n$  的所有因子。
- 样例输入  
8
- 样例输出  
1 2 4 8

## 例 7.3：输出 n 的所有因子

```
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         if (n % i == 0) {
10             cout << i << " ";
11         }
12     }
13     cout << endl;
14
15     return 0;
16 }
```

## 例 7.3：输出 n 的所有因子

```
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         if (n % i == 0) {
10             cout << i << " ";
11         }
12     }
13     cout << endl;
14
15     return 0;
16 }
```

## 例 7.4：求 $n$ 的所有因子之和

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 求  $n$  的所有因子之和。
- 样例输入  
8
- 样例输出  
15
- 样例说明  
 $1 + 2 + 4 + 8 = 15$

## 例 7.4：求 n 的所有因子之和

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

## 例 7.4：求 n 的所有因子之和

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



## 例 7.5: 求 $n$ 个数的最大值

### 编程题

- 第一行输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 第二行输入  $n$  个整数  $x$  ( $-10^9 \leq x \leq 10^9$ ), 求这  $n$  个整数中的最大值。
- 样例输入  
4  
8 7 9 5
- 样例输出  
9

## 例 7.5: 求 $n$ 个数的最大值

- 打擂台找最大值

8

7

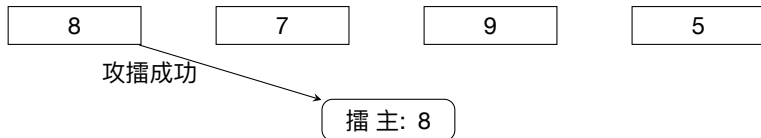
9

5

擂 主

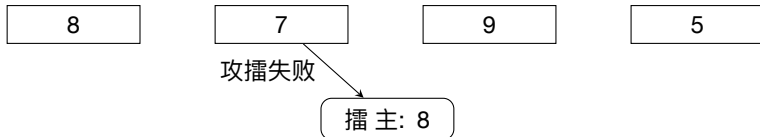
## 例 7.5: 求 n 个数的最大值

- 打擂台找最大值



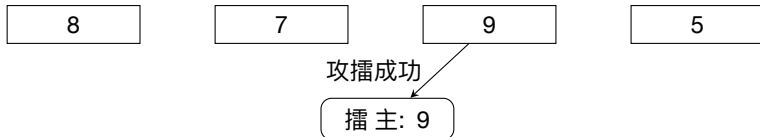
## 例 7.5: 求 n 个数的最大值

- 打擂台找最大值



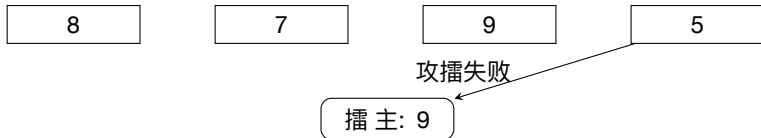
## 例 7.5: 求 n 个数的最大值

- 打擂台找最大值



## 例 7.5: 求 n 个数的最大值

- 打擂台找最大值



## 例 7.5：求 n 个数的最大值

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int maxv = -1000000000, x;
9     for (int i = 1; i <= n; i++) {
10         cin >> x;
11         if (x > maxv) {
12             maxv = x; // 攻擂成功，成为擂主
13         }
14     }
15     cout << maxv << endl;
16
17     return 0;
18 }
```

## 例 7.5: 求 n 个数的最大值

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int maxv = -1000000000, x;
9     for (int i = 1; i <= n; i++) {
10         cin >> x;
11         if (x > maxv) {
12             maxv = x; // 攻擂成功, 成为擂主
13         }
14     }
15     cout << maxv << endl;
16
17     return 0;
18 }
```

- 把擂主的初始值设置为最小值, 确保第一个挑战者攻擂成功



## 例 7.6：求 $n$ 个数的最大值及位置

### 编程题

- 第一行输入一个整数  $n$  ( $1 \leq n \leq 1000$ )，第二行输入  $n$  个整数  $x$  ( $-10^9 \leq x \leq 10^9$ )，求这  $n$  个整数中的最大值及位置（保证最大值是唯一的）。
- 样例输入  
4  
8 7 9 5
- 样例输出  
9 3

## 例 7.6：求 n 个数的最大值及位置

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     // maxv 记录最大值, pos 记录最大值是第几个数 (位置)
9     int maxv = -1000000000, pos = 0, x;
10    for (int i = 1; i <= n; i++) {
11        cin >> x;
12        if (x > maxv) {
13            maxv = x; // 攻擂成功, 成为擂主
14            pos = i;  // 记录擂主是当前第 i 个数
15        }
16    }
17    cout << maxv << " " << pos << endl;
18
19    return 0;
20 }
```

## 例 7.6：求 n 个数的最大值及位置

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     // maxv 记录最大值, pos 记录最大值是第几个数 (位置)
9     int maxv = -1000000000, pos = 0, x;
10    for (int i = 1; i <= n; i++) {
11        cin >> x;
12        if (x > maxv) {
13            maxv = x; // 攻擂成功, 成为擂主
14            pos = i; // 记录擂主是当前第 i 个数
15        }
16    }
17    cout << maxv << " " << pos << endl;
18
19    return 0;
20 }
```

# 目录

- 1 复习回顾
- 2 for 循环嵌套分支
- 3 while/do..while 循环嵌套分支**
- 4 break 和 continue 语句
- 5 总结

# 问题回顾：反向输出非负整数

## 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 10^9$ ), 请从低位到高位输出  $n$  的每个数位。
- 样例输入  
856
- 样例输出  
6 5 8

# 问题回顾：反向输出非负整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     do {
9         cout << n % 10 << " "; // 输出最低位
10        n /= 10;                // 抹去最低位
11    } while (n != 0);
12
13    return 0;
14 }
```

# 问题回顾：反向输出非负整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     do {
9         cout << n % 10 << " "; // 输出最低位
10        n /= 10;                // 抹去最低位
11    } while (n != 0);
12
13    return 0;
14 }
```

## 例 7.7: 统计数位中 3 的个数

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 10^9$ ), 统计  $n$  的数位中 3 的个数。
- 样例输入  
1353
- 样例输出  
2



## 例 7.7：统计数位中 3 的个数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int cnt = 0;
9     do {
10         if (n % 10 == 3) { // 判断取出的最低位是否为 3
11             cnt++;
12         }
13         n /= 10;           // 抹去最低位
14     } while (n != 0);
15     cout << cnt << endl;
16
17     return 0;
18 }
```

## 例 7.7: 统计数位中 3 的个数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int cnt = 0;
9     do {
10         if (n % 10 == 3) { // 判断取出的最低位是否为 3
11             cnt++;
12         }
13         n /= 10;           // 抹去最低位
14     } while (n != 0);
15     cout << cnt << endl;
16
17     return 0;
18 }
```

# 目录

- 1 复习回顾
- 2 for 循环嵌套分支
- 3 while/do..while 循环嵌套分支
- 4 break 和 continue 语句**
- 5 总结

# break 和 continue 语句

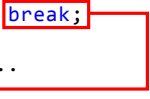
- break: 用于循环体中, 表示跳出本层循环

```
1 while (...) {  
2     ...  
3     if (...) {  
4         break;  
5     }  
6     ...  
7 }
```

# break 和 continue 语句

- break: 用于循环体中, 表示跳出本层循环

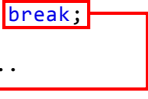
```
1 while (...) {  
2     ...  
3     if (...) {  
4         break;  
5     }  
6     ...  
7 }
```



# break 和 continue 语句

- break: 用于循环体中, 表示跳出本层循环

```
1 while (...) {  
2     ...  
3     if (...) {  
4         break;  
5     }  
6     ...  
7 }
```

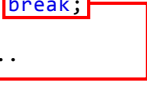


```
1 for (int i = 1; i <= n; i++) {  
2     ...  
3     if (...) {  
4         break;  
5     }  
6     ...  
7 }
```

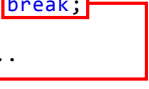
# break 和 continue 语句

- break: 用于循环体中, 表示跳出本层循环

```
1 while (...) {  
2     ...  
3     if (...) {  
4         break;  
5     }  
6     ...  
7 }
```



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



# break 和 continue 语句

- `continue`: 用于循环体中, 表示跳过本次循环 (循环体) 中余下未执行的语句, 进行下一次循环


```
1 while (...) {  
2     ...  
3     if (...) {  
4         continue;  
5     }  
6     ...  
7 }
```



# break 和 continue 语句

- **continue**: 用于循环体中，表示跳过本次循环（循环体）中余下未执行的语句，进行下一次循环


```
1 while (...) {  
2     ...  
3     if (...) {  
4         continue;  
5     }  
6     ...  
7 }
```



# break 和 continue 语句

- **continue**: 用于循环体中，表示跳过本次循环（循环体）中余下未执行的语句，进行下一次循环

```
1 while (...) {  
2     ...  
3     if (...) {  
4         continue;  
5     }  
6     ...  
7 }
```




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


# break 和 continue 语句

- **continue**: 用于循环体中，表示跳过本次循环（循环体）中余下未执行的语句，进行下一次循环

```
1 while (...) {  
2     ...  
3     if (...) {  
4         continue;  
5     }  
6     ...  
7 }
```



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



# break 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 break 语句, 观察输出结果。

# break 语句 - 示例

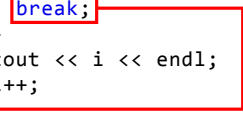
- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 break 语句, 观察输出结果。

```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         break;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```

# break 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 break 语句, 观察输出结果。

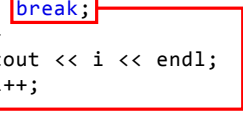
```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         break;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```



# break 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 break 语句, 观察输出结果。

```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         break;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```



```
1 int n;  
2 cin >> n;  
3 for (int i = 1; i <= n; i++) {  
4     if (i == 5) {  
5         break;  
6     }  
7     cout << i << endl;  
8 }
```

# break 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 break 语句, 观察输出结果。

```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         break;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```

```
1 int n;  
2 cin >> n;  
3 for (int i = 1; i <= n; i++) {  
4     if (i == 5) {  
5         break;  
6     }  
7     cout << i << endl;  
8 }
```



# continue 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 `continue` 语句, 观察输出结果。

# continue 语句 - 示例


- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 continue 语句, 观察输出结果。

```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         continue;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```

# continue 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 `continue` 语句, 观察输出结果。


```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         continue;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```



# continue 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 continue 语句, 观察输出结果。

```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         continue;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```




```
1 int n;  
2 cin >> n;  
3 for (int i = 1; i <= n; i++) {  
4     if (i == 5) {  
5         continue;  
6     }  
7     cout << i << endl;  
8 }
```


# continue 语句 - 示例

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出  $1 \sim n$  中的每个整数, 增加判断当这个数为 5 时, 使用 continue 语句, 观察输出结果。

```
1 int n;  
2 cin >> n;  
3 int i = 1;  
4 while (i <= n) {  
5     if (i == 5) {  
6         continue;  
7     }  
8     cout << i << endl;  
9     i++;  
10 }
```



```
1 int n;  
2 cin >> n;  
3 for (int i = 1; i <= n; i++) {  
4     if (i == 5) {  
5         continue;  
6     }  
7     cout << i << endl;  
8 }
```



# break 和 continue 语句

- break
  - 在 while 循环中，直接跳出本层 while 循环
  - 在 for 循环中，直接跳出本层 for 循环

# break 和 continue 语句

- break
  - 在 while 循环中，直接跳出本层 while 循环
  - 在 for 循环中，直接跳出本层 for 循环
- continue
  - 在 while 循环中，直接跳转到**条件语句**，需要额外注意
  - 在 for 循环中，直接跳转到每次循环后的动作

## 填空题

### 1. 阅读程序写结果

```
1 // #include ...
2
3 int main() {
4     int a, b;
5     cin >> a >> b;
6     for (int i = a; i <= b; i++) {
7         if (i % 5 == 0) {
8             break;
9         }
10        cout << i << " ";
11    }
12    cout << endl;
13
14    return 0;
15 }
```

输入: 7 12

输出:



## 填空题

### 1. 阅读程序写结果

```
1 // #include ...
2
3 int main() {
4     int a, b;
5     cin >> a >> b;
6     for (int i = a; i <= b; i++) {
7         if (i % 5 == 0) {
8             break;
9         }
10        cout << i << " ";
11    }
12    cout << endl;
13
14    return 0;
15 }
```

输入: 7 12

输出: 7 8 9

# 目录

- 1 复习回顾
- 2 for 循环嵌套分支
- 3 while/do..while 循环嵌套分支
- 4 break 和 continue 语句
- 5 总结

- 循环与分支综合
  - 求和问题
  - 计数问题
  - 最值问题
  - 数位拆分
- break 和 continue 语句

# Thank you!