

第三章 循环结构

第 3 2 课 循环控制语句

《信息学奥赛一本通·编程启蒙 C++版》

一、无限循环

死循环（endless loop）是指在编程中，一个靠自身控制无法终止的程序。例如在 C++ 语言程序中，语句 “`while(1)printf("*");`” 就是一个死循环，运行它将无休止地打印*号。

死循环的 C++ 语言实现：

(1) `while(1);`//1 表示循环条件永远成立

(2) `for(;;);`//跳出循环的条件为空，那么循环将会被永远执行

在循环结构程序中，有时需要提前终止循环，或者跳过特定的语句，或退出当前的函数，这时可以用 `break` 语句、`continue` 语句或 `return` 语句来实现。

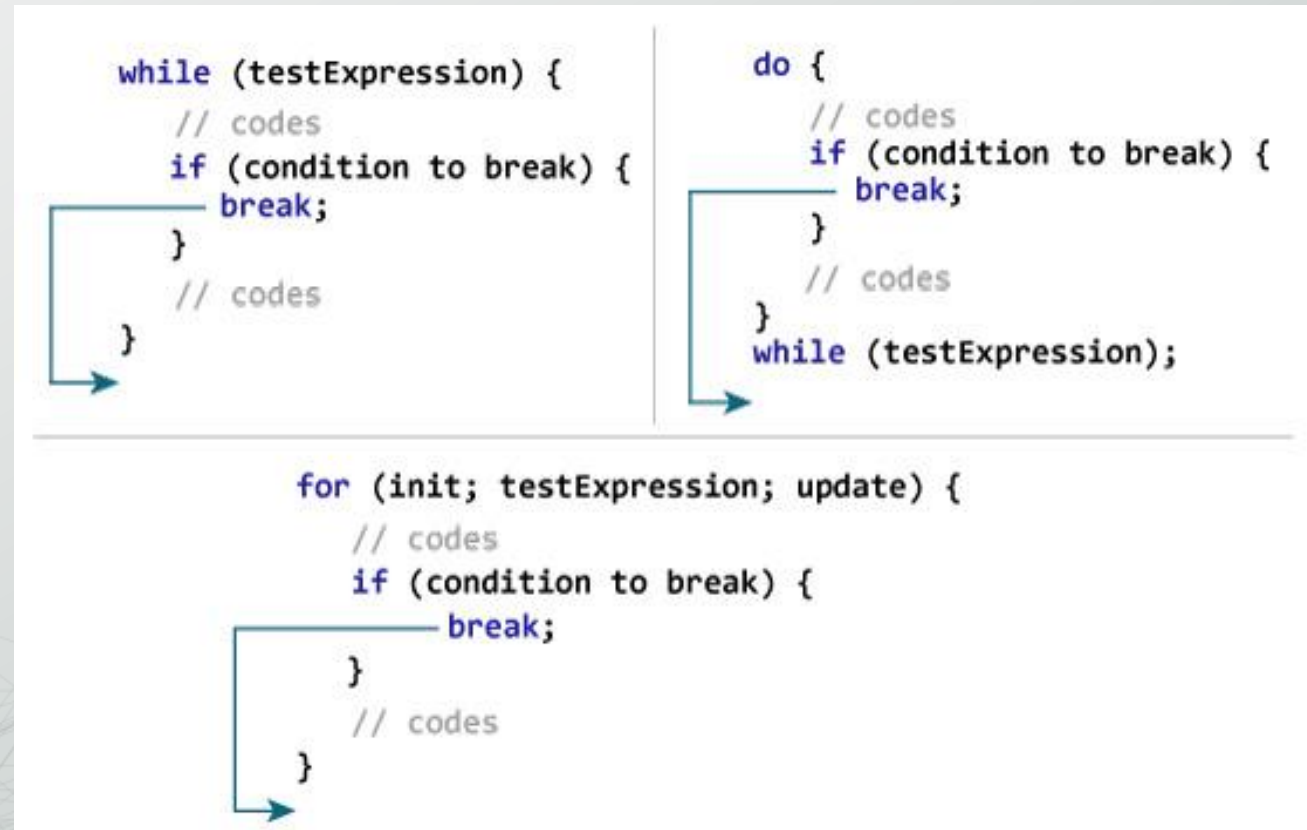
二、break 语句

C++语言中 **break** 语句有以下两种用法：

(1) 当 **break** 语句出现在一个循环内时，循环会立即终止，且程序流将继续执行紧接着循环的下一条语句。

(2) 它可用于终止 **switch** 语句中的一个 **case**。

如果您使用的是嵌套循环（即一个循环内嵌套另一个循环），**break** 语句会停止执行最内层的循环，然后开始执行该块之后的下一行代码。



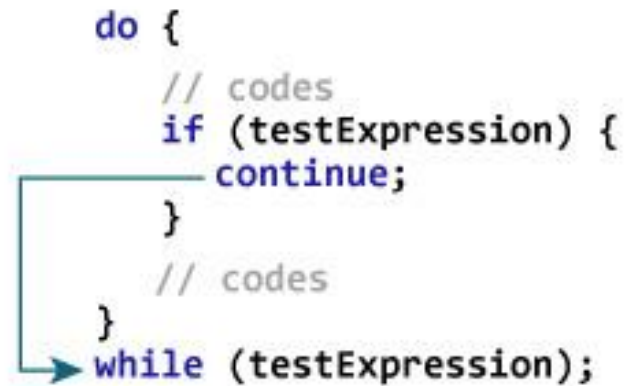
三、continue 语句

C++ 中的 `continue` 语句有点像 `break` 语句。但它不是强迫终止，`continue` 会跳过当前循环中的代码，强迫开始下一次循环。

对于 `for` 循环，`continue` 语句会导致执行条件测试和循环增量部分。对于 `while` 和 `do...while` 循环，`continue` 语句会导致程序控制回到条件测试上。



```
while (testExpression) {  
    // codes  
    if (testExpression) {  
        continue;  
    }  
    // codes  
}
```



```
do {  
    // codes  
    if (testExpression) {  
        continue;  
    }  
    // codes  
} while (testExpression);
```



```
for (init; testExpression; update) {  
    // codes  
    if (testExpression) {  
        continue;  
    }  
    // codes  
}
```

四、使用 `return` 退出函数

`return 0`，说明程序正常退出。

`return` 是 C++ 预定义的语句，它提供了终止函数执行的一种方式。当 `return` 语句提供了一个值时，这个值就成为函数的返回值。

【例 32.1】 学游泳

【题目描述】

小严开心地在游泳，可是他很快难过地发现，自己的力气不够，游泳好累哦。已知小严第一步能游 2 米，可是随着越来越累，力气越来越小，小严接下来的每一步都只能游出上一步距离的 98%。现在小严想知道，如果要游到距离 x 米的地方，他需要游多少步呢。请你编程解决这个问题。

【输入格式】

输入一个实数 $x(1 \leq x \leq 99.9)$ ，表示要游的目标距离。

【输出格式】

输出一个整数，表示小严一共需要游多少步

【样例输入】

5.89

【样例输出】

4

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. double n,o=2,cnt;
4. int main(){
5.     cin>>n;
6.     for(int i=1;;i++){
7.         if(n<=0) break;
8.         n-=o;
9.         o*=0.98;
10.        cnt++;
11.    }
12.    cout<<cnt;
13.    return 0;
14.}
```

【例 32.2】 凑数

【题目描述】

现有两个正整数 x, y ，试问你在 1000 以内最大既不是 x 的倍数，也不是 y 的倍数的正整数是多少？

【输入格式】

只有两个整数 $x, y (2 \leq x, y \leq 1000)$ 。

【输出格式】

一个正整数表示答案。

【样例输入】

2 3

【样例输出】

997

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,m;
4. int main()
5. {
6.     cin>>n>>m;
7.     for(int i=1000;i>=0;i--){
8.         if(i%n!=0&& i%m!=0){
9.             cout<<i;
10.            return 0;
11.        }
12.    }
13.}
```

【例 32.2】 数位积

【题目描述】

× 【输入格式】

一开始有一个数字 $T(1 \leq T \leq 100)$ ，表示共有几组测试数据。接下来有 T 个数字 $n(0 \leq n < 2147483648)$ 。

【输出格式】

输出数位乘积的结果。

【样例输入】

3

356

123

9999

【样例输出】

90

6

6561

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,j,x;
4. int main(){
5.     cin>>n;
6.     for(int i=1;i<=n;i++){
7.         cin>>x;
8.         if(x==0) {
9.             cout<<0<<endl;
10.            continue; //跳出for 循环一次
11.        }
12.        j=1;
13.        while(x){
14.            j*=x%10;
15.            x/=10;
16.            if(j==0) break;//跳出while 这层的循环
17.        }
18.        cout<<j<<endl;
19.    }
20.    return 0;
21.}
```

练 32.1 立方和不等式

【题目描述】

试求满足下述立方和不等式的 m 的最大整数解 $1^3 + 2^3 + \cdots + m^3 \leq n$ 。

【输入格式】

输入 $n(1 \leq n \leq 10^5)$ 的值。

【输出格式】

输出最大的 m 。

【样例输入】

50

【样例输出】

3

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int sum;
4. int main (){
5.     cin>>sum;
6.     for(int i=1; ;i++){
7.         if(i*i*i<=sum)
8.             sum-=i*i*i;
9.         else{
10.             cout<<i-1;
11.             break;
12.         }
13.     }
14.     return 0;
15. }
```

练 32.2 种植竹笋

【题目描述】

竹笋的初始高度为 0，它的生长速度会随着天数逐渐递减：第一天长 16 厘米，第二天长 8 厘米……第 n 天长 $\frac{16}{n}$ 厘米（竹笋每天生长的高度可能不为整数）。竹笋长到 k 厘米及以上就可以收获然后售卖了，那么农民大叔在第几天结束后，就可以收获这些竹笋了？

如果农民大叔等了一年之后（第 365 天结束后），竹笋仍然不能达到收获高度，就输出 -1。

【输入格式】

输入共一行，为一个整数 k ($1 \leq k \leq 120$)，表示竹笋可以收获时的最低高度。

【输出格式】

输出共一行，为一个整数，表示第几天结束后可以收获竹笋。

【样例输入】

10

【样例输出】

1

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int k;
4. double h;
5. int main(){
6.     cin>>k;
7.     for(int i=1;i<=365;i++){
8.         h+=16.0/i;
9.         if(h>=k){
10.             cout<<i;
11.             return 0;
12.         }
13.     }
14.     cout<<-1;
15.     return 0;
16. }
```

练 32.3 买房子

【题目描述】

小严从现在开始工作，年薪 N 万。他希望在宁波附近买一套 60 平米的房子，现在价格是 200 万。假设房子价格以每年百分之 K 增长，并且小严未来年薪不变，且不吃不喝，不用交税，每年所得 N 万全都积攒起来，问第几年能够买下这套房子？（第一年年薪 N 万，房价 200 万）。

【输入格式】

一行，包含两个正整数 $N(10 \leq N \leq 50)$ ， $K(1 \leq K \leq 20)$ ，中间用单个空格隔开。

【输出格式】

如果在第 20 年或者之前就能买下这套房子，则输出一个整数 M ，表示最早需要在第 M 年能买下；否则输出 "*Impossible*"。

【样例输入】

50 10

【样例输出】

8

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. double sum,n,k,f=200;
4. int main(){
5.     cin>>n>>k;
6.     sum=n;
7.     if(sum>=200){
8.         cout<<1;
9.         return 0;
10.    }
11.    for(int i=2;i<=20;i++){
12.        f=f*(1+k/100);
13.        sum+=n;
14.        if(sum>=f){
15.            cout<<i;
16.            return 0;
17.        }
18.    }
19.    cout<<"Impossible";
20.    return 0;
21. }
```

谢谢！