

第三章 循环结构

第 3 9 课 时间复杂度

《信息学奥赛一本通·编程启蒙 C++版》

表示一个算法所编程序运行时间的多少，用的并不是准确值，而是根据合理方法得到的预估值。不同程序的运行时间，我们将在最坏条件下程序的运行时间作为预估值。数据结构推出了大 O 记法（注意，是大写的字母 O ，不是数字 0 ）来表示算法（程序）的运行时间。

```
1. {  
2.     a++;  
3. }
```

线性的代码

```
1. for(int i = 0 ; i < n ; i++) //<- 从 0 到 n，执行 n+1 次  
2. {  
3.     a++; //<- 从 0 到 n-1，执行 n 次  
4. }
```

单重循环

```
1. for(int i = 0 ; i < n ; i++)          // n+1  
2. {  
3.     for(int j = 0 ; j < m ; j++)      // n*(m+1)  
4.     {  
5.         num++;                        // n*m  
6.     }  
7. }
```

双重循环

例如，用大 O 记法表示上面 3 段程序的运行时间，则上面第一段程序的时间复杂度为 $O(1)$ ，第二段程序的时间复杂度为 $O(n)$ ，第三段程序的时间复杂度为 $O(n^2)$ 。

如下列举了常用的几种时间复杂度，以及它们之间的大小关系：

$$O(1) < O(\log n) < O(n) < O(n^2) < O(n^3) < O(2^n)$$

常数阶 < 对数阶 < 线性阶 < 平方阶 < 立方阶 < 指数阶

【例 39.1】 鼓掌次数

【题目描述】

幼稚园的小朋友们排成一列做游戏，他们从 1 开始报数，今天的幸运数字是 k ，报 k 的倍数的小朋友需要鼓掌一次，那么他们从头报数到尾，一共会鼓掌几次？

【输入格式】

一行两个整数 $n, k (1 \leq n, k \leq 2 \times 10^9)$ ，表示小朋友的人数和幸运数字。对于 90% 的数据， $1 \leq n, k \leq 10^6$ 。

对于 100% 的数据， $1 \leq n, k \leq 2 \times 10^9$ 。

【输出格式】

一行一个整数，表示鼓掌的次数。

【样例输入】

10 2

【样例输出】

5

【代码实现 1】

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

【代码实现 2】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,k;
4. int main(){
5.     cin>>n>>k;
6.     cout<<n/k;
7.     return 0;
8. }
```

【例 39.2】 球弹跳高度的计算

【题目描述】

一个球从某一高度落下（整数，单位米），每次落地后反跳回原来高度的一半，再落下。编程计算气球在第 10 次落地时，共经过多少米？第 10 次反弹多高？

【输入格式】

输入一个整数 h ($1 \leq h \leq 100$)，表示球的初始高度。

【输出格式】

输出包含两行：

第 1 行：到球第 10 次落地时，一共经过的米数。

第 2 行：第 10 次弹跳的高度。

【样例输入】

18

【样例输出】

53.9297

0.0175781

【代码实现 1】

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

```
}
```

【代码实现 2】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. double n;
4. int main()
5. {
6.     cin>>n;
7.     cout<<2*n+255*n/256<<endl;
8.     cout<<n/1024<<endl;
9.     return 0;
10. }
```

练 39.1 书香阁座位 (趣味编程)

【题目描述】

风之巅小学的书香阁有 **312** 个座位，已知第一排有 **15** 个座位，以后每排比上一排增加 **2** 个座位，最后一排有几个座位？一共有几排？

试编一程序算一算。

【输入格式】

无。

【输出格式】

对于倒数第二行，输出最后一排的座位数。

对于最后一行，输出排数。

【样例输入】

无

【样例输出】

最后一排的座位数： 37

排数： 12

【代码实现】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int sum,p,x;
6.     p=1;
7.     x=15;
8.     sum=x;
9.     //cout<<p<<" "<<x;  //输出第一排的座位信息
10.    //cout<<" "<<sum<<endl;
11.    do
12.    {
13.        p++;
14.        x+=2;
15.        sum+=x;
16.    //cout<<p<<" "<<x<<" "<<sum<<endl; //输出当前排的座位信息
17.    }while(sum!=312);
18.    cout<<"最后一排的座位数: "<<x<<endl; //输出最后一排的座位数
    和排数
19.    cout<<"排数: "<<p<<endl;
20.    return 0;
21. }
```

练 39.2 蜗牛爬树 (趣味编程)

【题目描述】

有一棵光滑的葡萄树高 **17** 分米，一只蜗牛从底部向上爬，每分钟爬 **3** 分米，但每爬一分钟后都要休息一分钟，休息期间又要滑下 **1** 分米。该蜗牛需要多少分钟才能爬到树顶？

【输入格式】

无。

【输出格式】

输出一个整数，表示蜗牛需要多少分钟才能爬到树顶。

【样例输入】

无

【样例输出】

需要 **15** 分钟

【代码实现】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int i,t; //i 表示蜗牛爬的高度, t 表示分钟数。
6.     t=i=0;
7.     while(1) //若在循环体中没有执行break 语句终止循环, 则会形成
        死循环。
8.     {
9.         t++;
10.        i+=3;
11.        if(i>=17)break; //若爬到树顶, 则终止循环。
12.        t++;
13.        i--;
14.    }
15.    cout<<"需要"<<t<<"分钟"<<endl;
16.}
```

练 39.3 鸡兔同笼 (趣味编程)

【题目描述】

《孙子算经》记载：“今有雉兔同笼，上有三十五头，下有九十四足，问雉兔各几何？”有若干只鸡同在一个笼子里，从上面数，有 35 个头，从下面数，有 94 只脚，问鸡和兔各有几只。

【输入格式】

无。

【输出格式】

一行，分别输出鸡和兔的数量。

【样例输入】

无

【样例输出】

鸡： 23 兔： 12

【代码实现】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int ji,tu;
6.     for(ji=1;ji<=34;ji++)// 枚举鸡
7.     for(tu=1;tu<=23;tu++)// 枚举兔
8.     if(ji+tu==35&&ji*2+tu*4==94)// 判断
9.         cout<<"鸡: "<<ji<<" 兔: "<<tu<<endl;
10.    return 0;
11.}
```

练 39.4 百钱买百鸡 (趣味编程)

【题目描述】

百钱买百鸡问题：3 文钱可以买 1 只公鸡，2 文钱可以买 1 只母鸡，1 文钱可以买 3 只小鸡，要用 100 文钱买 100 只鸡，求公鸡，母鸡，小鸡各多少只？

【输入格式】

无。

【输出格式】

第一行分别输出公鸡，母鸡，小鸡，按每个词语占 5 个字符的宽度。接着输出各种公鸡、母鸡、小鸡的数量，依次由小到大，每种情况各占一行，并每个整数占 5 个字符的宽度。

【样例输入】

无

【样例输出】

公鸡	母鸡	小鸡
5	32	63
10	24	66
15	16	69
20	8	72

【代码实现】

```
1. #include<iostream>
2. #include<iomanip>
3. using namespace std;
4. int main()
5. {
6.     int gongji,muji,xiaoji;
7.     cout<<"    "<<"公鸡"<<"    "<<"母鸡"<<"    "<<"小鸡"<<endl;
8.     for(gongji=1;gongji<=33;gongji++) //枚举公鸡
9.     for(muji=1;muji<=50;muji++) //枚举母鸡
10.    {
11.        xiaoji=100-gongji-muji; //枚举小鸡
12.        if(gongji*3+muji*2+xiaoji/3.0==100) //判断
13.        cout<<setw(5)<<gongji<<setw(5)<<muji<<setw(5)<<xiao
        ji<<endl;
14.    }
15.    return 0;
16.}
```

谢谢！