



输出偶数

【问题描述】

输出 1~100 之间所有偶数。

【输出结果】

```
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78
80 82 84 86 88 90 92 94 96 98 100
```

【方法 1】

对于 1~100 之间的 100 个数字 i ，直接循环进行判断，如果 i 是偶数，则输出 i 的值。

这种思想，本质上是穷举。穷举法保证在求解的过程中，所有可能的解都会判断到，不会丢解。

穷举的缺点是效率不高。

【方法 1】参考程序

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     //对于 1 至 100 之间的每个数，都重复操作
6.     for (int i=1; i<=100; i++)
7.     {
8.         //如果 i 为偶数，则输出 i 的值
9.         if (i%2==0)
10.        {
11.            cout << i << " ";
12.        }
13.    }
14.    return 0;
15. }
```

【方法 2】

相邻偶数之间的差值为 2，所以，可以设置 for 循环的循环变量的初值为 2，增量为 2，使得循环次数减少为 50 次。

**【方法 2】参考程序**

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     //i 初值 2，终值 100，每次增量为 2
6.     for (int i=2; i<=100; i+=2)
7.     {
8.         cout << i << " ";
9.     }
10.    return 0;
11. }
```



累加器

【问题描述】

利用 for 循环，计算 $1 + 2 + \dots + 100$ 的和，并输出。

【输出结果】

5050

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     //总和
6.     int sum = 0;
7.     //i 初始值为 1，终值为 100，每次增量为 1
8.     for (int i=1; i<=100; i++)
9.     {
10.         sum += i;
11.     }
12.     cout << sum;
13.     return 0;
14. }
```

【分析】

可以参照下面表格理解循环累加是如何完成的。

	sum	i
第 1 次运行前	0	1
第 1 次运行后	0+1	2
第 2 次运行后	0+1+2	3
第 3 次运行后	0+1+2+3	4
...		



第 99 次运行后	$0+1+2+3+\cdots+99$	100
第 100 次运行后	$0+1+2+3+\cdots+100$	101



阶乘

【问题描述】

输入正整数 n ($n \leq 20$)，利用 for 循环计算 $n!$ 的值。

【分析】

$$n! = 1 \times 2 \times 3 \times \dots \times n$$

【输入样例】

13

【输出样例】

6227020800

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     //若 s 定义为 int，当 n=13 时 s 的值就溢出了
6.     long long s = 1;
7.     int n;
8.     cin >> n;
9.     for (int i=1; i<=n; i++)
10.    {
11.        s *= i;
12.    }
13.    cout << s << endl;
14.    return 0;
15. }
```



偶数和奇数的和

【问题描述】

利用 for 循环，分别计算 1~100 中奇数的和、偶数的和。

【输出结果】

2550 2500

【方法 1】

用变量 sum1 和 sum2 分别存放偶数与奇数和。新来一个数字，如果是偶数就累加 sum1 上，如果是奇数就累加在 sum2 上。

【方法 1】参考程序

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int sum1 = 0; //偶数和
6.     int sum2 = 0; //奇数和
7.     //i 取 1 至 100 之间的每个整数，都重复操作
8.     for (int i=1; i<=100; i++)
9.     {
10.        if (i%2==0)
11.        {
12.            //偶数累加到 sum1 中
13.            sum1 += i;
14.        }
15.        else
16.        {
17.            //奇数累加到 sum2 中
18.            sum2 += i;
19.        }
20.    }
21.    //输出偶数和、奇数和
22.    cout << sum1 << " " << sum2;
23.    return 0;
24. }
```



【方法 2】

偶数从 2 开始每次递增 2，奇数从 1 开始每次递增 2，for 语句的循环变量初始化和循环变量更新两部分都可以使用逗号语句序列。

【方法 2】参考程序

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int i, j, sum1=0, sum2=0;
6.     //生成偶数和奇数的初始值
7.     for (i=2, j=1; i<=100; i+=2, j+=2)
8.     {
9.         sum1+=i;    //偶数 i 累加到 sum1 中
10.        sum2+=j;    //奇数 j 累加到 sum2 中
11.    }
12.    //输出偶数和、奇数和
13.    cout << sum1 << " " << sum2;
14.    return 0;
15. }
```



数据统计

【问题描述】

输入一些整数，求出它们的最小值、最大值和平均值。

【输入样例】

```
8
2 8 3 5 1 7 3 6
```

【输出样例】

```
1 8 4.375
```

【参考程序 1】

下面程序输出的结果是不是正确的呢？

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int n, min, max, s=0;
6.     cin >> n;
7.     for(int i=0; i<n; i++)
8.     {
9.         int x;
10.        cin >> x;
11.        s += x;
12.        if (x < min)
13.        {
14.            min = x;
15.        }
16.        if (x > max)
17.        {
18.            max = x;
19.        }
20.    }
21.    cout << min << ' ' << max << ' ' << (double)s/n << endl;
22.    return 0;
23. }
```




是不正确的。因为 min 和 max 并没有初始化初值。也就是说最开始的 min 和 max 的取值是随机不确定的。

【参考程序 2】

那下面的程序是不是就没有问题了？

```
1. #include<iostream>
2. using namespace std;
3. const int INF = 100000000;
4. int main()
5. {
6.     int n, min=INF, max=-INF, s=0;
7.     cin >> n;
8.     for(int i=0; i<n; i++)
9.     {
10.         int x;
11.         cin >> x;
12.         s += x;
13.         if (x < min)
14.         {
15.             min = x;
16.         }
17.         if (x > max)
18.         {
19.             max = x;
20.         }
21.     }
22.     cout << min << ' ' << max << ' ' << (double)s/n << endl;
23.     return 0;
24. }
```

事实上这个程序还是可能存在问题的。一种可能是 INF 定义的值不够大。还有一种可能是 INF 定义的值过大，导致程序溢出了。

【参考程序 3】

那最完美的程序是什么样子的呢？

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int n, x;
```



```
6.     cin >> n;
7.     cin >> x;
8.     int s = x;
9.     int min = x;
10.    int max = x;
11.    for(int i=1; i<n; i++)
12.    {
13.        cin >> x;
14.        s += x;
15.        if (x < min)
16.        {
17.            min = x;
18.        }
19.        if (x > max)
20.        {
21.            max = x;
22.        }
23.    }
24.    cout << min << ' ' << max << ' ' << (double)s/n << endl;
25.    return 0;
26. }
```

只需要先读取一个数，把它赋值给 min 和 max，其实就可以了。