



房间的门

【问题描述】

宾馆里有 N (最大 1000) 个房间, 从 1~ N 编号。第一个服务员把所有的房间门都打开了, 第二个服务员把所有编号是 2 的倍数的房间“相反处理”, 第三个服务员把所有编号是 3 的倍数的房间作“相反处理”..., 以后每个服务员都是如此。当第 N 个服务员来过后, 哪几扇门是打开的。

“相反处理”指的是: 原来开着的门关上, 原来关上的门打开。

【输入样例】

100

【输出样例】

1 4 9 16 25 36 49 64 81 100

【分析】

用 $a[1], a[2], \dots, a[n]$ 表示编号为 1, 2, 3, ..., n 的门是否开着。模拟服务员的操作即可。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. const int N = 1000;
4. int main()
5. {
6.     bool a[N+1] = {};
7.     int n;
8.     cin >> n;
9.     //依次循环服务员
10.    for (int i=1; i<=n; i++)
11.    {
12.        //i 号服务员依次处理每个门
13.        for (int j=1; j<=n; j++)
14.        {
15.            if (j%i == 0)
16.            {
17.                a[j] = !a[j];
```



```
18.         }
19.     }
20. }
21. for (int i=1; i<=n; i++)
22. {
23.     if (a[i])
24.     {
25.         cout << i << " ";
26.     }
27. }
28. return 0;
29. }
```



筛法求素数

【问题描述】

用筛法 (Sieve Method) 求出 N (最大 1000) 以内的全部素数。

用筛法求素数的过程示意如下：

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15...98 99 100 //置数
- 2 3 4 5 6 7 8 9 10 11 12 13 14 15...98 99 100 //筛去除 2 以外被 2 整除的数
- 2 3 4 5 6 7 8 9 10 11 12 13 14 15...98 99 100 //筛去除 3 以外被 3 整除的数
-
- 2 3 4 5 6 7 8 9 10 11 12 13 14 15...98 99 100 //筛去被 p 整除的数

【输入样例】

100

【输出样例】

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83
89 97

【分析】

- 1) 布尔数组 a 记录每个自然数是否被划去；初始从 2 到 n，都没有被划去。
- 2) 在数组元素中，以下标为序，按顺序找到未被划去的最小素数 i。
- 3) 把 n 以内 i 的倍数从 a 数组中全部划去，也就是给该元素值置 false。
- 4) 重复执行第 (2)、(3) 步骤，直到 $i > \sqrt{n}$ 为止。
- 5) 输出数组 a 中未被筛掉的各个元素。

【参考程序】

```
1. #include<iostream>
2. #include<cmath>
```



```
3. using namespace std;
4. const int N = 1000;
5. int main()
6. {
7.     bool a[N+1] = {};
8.     int n;
9.     cin >> n;
10.    for (int i=2; i<=n; i++)
11.    {
12.        a[i] = true;
13.    }
14.    for (int i=2; i<=sqrt(n); i++)
15.    {
16.        if (a[i])
17.        {
18.            for (int j=2; j<=n/i; j++)
19.            {
20.                a[i*j] = false;
21.            }
22.        }
23.    }
24.    for (int i=2; i<=n; i++)
25.    {
26.        if (a[i])
27.        {
28.            cout << i << " ";
29.        }
30.    }
31.    return 0;
32. }
```