

☑ PAT 乙级题目讲解：1013 《数素数》

🔗 题目简介

本题要求输出第 P_M 到第 P_N 个素数，其中 P_i 表示第 i 个素数。

输出格式为： 每行最多输出 10 个素数，素数之间用空格隔开，**末尾不得多输出空格或换行。**

🔧 样例分析

输入：

```
5 27
```

前 27 个素数依次为：

```
2 3 5 7 11 13 17 19 23 29
31 37 41 43 47 53 59 61 67 71
73 79 83 89 97 101 103
```

我们需要输出从第 5 个素数（即 11）到第 27 个素数（即 103）之间的所有素数，共 23 个。

输出格式要求：

- 每行最多输出 10 个素数；
- 素数之间用空格隔开；
- 最后一行末尾不能有多余空格。

输出：

```
11 13 17 19 23 29 31 37 41 43
47 53 59 61 67 71 73 79 83 89
97 101 103
```

🔍 解题思路

本题是典型的素数筛选 + 输出格式控制问题。

🔗 变量说明

变量名	含义
<code>maxn</code>	筛法范围上限 ($10^6 + 5$)
<code>a[i]</code>	素数筛标记, 0 表示是素数, 1 表示是合数
<code>b[i]</code>	存储前若干个素数, 第 i 个素数为 <code>b[i]</code>
<code>m, n</code>	题目给定的 M, N , 输出第 m 到第 n 个素数
<code>k</code>	当前已经找到的素数个数, 用于填充 <code>b</code> 数组
<code>c</code>	当前已经输出了多少个素数, 用于换行控制

本题的解决流程可以分为以下几个步骤：

☑ Step 1. 筛选素数（埃拉托色尼筛法）

我们使用 **埃拉托色尼筛法** 预处理一定范围内的素数：

- 设置最大范围 `maxn = 1e6 + 5`，保证可以筛出前 10000 个素数；
- `a[i] == 0` 表示 i 是素数；
- 从 $i = 2$ 开始，标记 i 的所有倍数为合数。

```
const int maxn = 1e6 + 5;
bool a[maxn]; // a[i] == 0 表示 i 是素数

// 筛选素数（埃拉托色尼筛法）
for(int i = 2; i * i <= maxn; i++){
    if(!a[i]){
        for(int j = 2 * i; j <= maxn; j += i){
            a[j] = 1; // 筛掉合数
        }
    }
}
```

☑ Step 2. 提取前 10000 个素数

定义一个 `b` 数组用于存储前 10000 个素数（即 P_1 到 P_{10000} ）：

- 遍历筛选数组 `a`；
- 将素数依次填入 `b` 数组；
- 一旦素数数量达到 10000 就停止。

```
int k = 0;
for(int i = 2; i <= maxn; i++){ // 提取前10000个素数
    if(!a[i]){
        b[++k] = i;
        if(k > 10000) break;
    }
}
```

☑ Step 3. 输出第 P_M 到第 P_N 个素数，并控制格式

设变量 `c` 记录当前输出的素数数量：

- 从 $b[m]$ 输出到 $b[n]$ ；
- 每输出一个数，`c++`；
- 每满 10 个数字输出换行；
- 最后一个数字后不输出空格或换行符，需特判。

```
int c = 0;
for(int i = m; i <= n; i++){
    cout << b[i];
    c++; // 计数已输出数字个数
    if(i == n) continue; // 最后一个数字后不加空格或换行
    if(c % 10 == 0) cout << "\n"; // 每 10 个换行
    else cout << " ";
}
```

☑ 完整代码

```
#include <bits/stdc++.h>
using namespace std;

const int maxn = 1e6 + 5;
bool a[maxn]; // a[i] == 0 表示 i 是素数
int m, n, b[10005], k;

int main(){
    cin >> m >> n;

    // 筛选素数（埃拉托色尼筛法）
    for(int i = 2; i * i <= maxn; i++){
        if(!a[i]){
            for(int j = 2 * i; j <= maxn; j += i){
                a[j] = 1; // 筛掉合数
            }
        }
    }

    // 提取前10000个素数
    for(int i = 2; i <= maxn; i++){
        if(!a[i]){
            b[++k] = i;
            if(k > 10000) break;
        }
    }

    // 输出格式控制
    int c = 0;
    for(int i = m; i <= n; i++){
        cout << b[i];
        c++;
        if(i == n) continue; // 最后一个数字后不加空格或换行
        if(c % 10 == 0) cout << "\n";
        else cout << " ";
    }

    return 0;
}
```

```
}
```

🚩 常见错误提醒

错误类型	具体表现
输出格式错误	每 10 个数后未换行或最后一个数后输出空格
数组越界	<code>b[i]</code> 下标超出范围，找到第 10000 个素数就要 break 停止
素数预处理不足	<code>maxn</code> 太小找不到足够素数

☑️ 总结归纳

- 本题本质是素数筛选 + 输出格式控制；
- 使用埃拉托色尼筛法，高效筛选前 10^4 个素数；
- 注意从第 P_m 个开始计数，不是从 m 本身；
- **时间复杂度：** $O(n \log \log n)$
- **空间复杂度：** $O(n)$ ，主要用于布尔筛选数组。

🧠 思维拓展

- 如果范围更大，可考虑 **线性筛法**，复杂度 $O(n)$ ；
- 你也可以尝试用 `isPrime()` 函数暴力判断，但效率远低；
- 输出格式控制是算法题常考点，建议写个通用模板练习。