

题目描述

输入 n ($n \leq 10^6$) 个不超过 10^9 的单调不减的（就是后面的数字不小于前面的数字）非负整数 a_1, a_2, \dots, a_n 然后进行 m ($m \leq 105$) 次询问。对于每次询问，给出一个整数 q ($q \leq 10^9$)，要求输出这个数字在序列中第一次出现的编号，如果没有找到的话输出 -1 。

输入格式

第一行 2 个整数 n 和 m ，表示数字个数和询问次数。

第二行 n 个整数，表示这些待查询的数字。

第三行 m 个整数，表示询问这些数字的编号，从 1 开始编号。

输出格式

m 个整数表示答案。

输入样例

```
11 3
1 3 3 3 5 7 9 11 13 15 15
1 3 6
```

输出样例

```
1 2 -1
```

解析

很明显，这是一道边界搜索，直接套用公式代码。

编码

```
#include <bits/stdc++.h>

using namespace std;

//定义全局最大数
const int MaxNum = 1e6 + 1;

//nums 待搜索数组
//target 目标数字
//length 数组长度
//return 目标所在索引位置
int BinarySearchLeft(int nums[], int target, int length) {
    int left = 1; //左边界
```

```

//我们将长度进行了减1
int right = length - 1;
//开始循环搜索
while (left <= right) {
    int mid = left + (right - left) / 2;
    if (nums[mid] == target) {
        //持续向左搜索
        right = mid - 1;
    }
    //小于目标，边界右移
    else if (nums[mid] < target) {
        left = mid + 1;
    }
    //大于目标，边界左移
    else if (nums[mid] > target) {
        right = mid - 1;
    }
}
//寻找的目标数字不存在于当前数组
if (left >= length || nums[left] != target) {
    return -1;
}
return left;
}

int n, m; //数组长度和询问次数
int a[MaxNum];
int main() {
    //输入初始数据
    cin >> n >> m;
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
    }
    //循环调用边界搜索
    for (int i = 0; i < m; ++i) {
        int target;
        cin >> target;
        cout << BinarySearchLeft(a, target, n) << " ";
    }
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

