

题目描述：

某个产品当前迭代周期内有 N 个特性 ($\{F_1, F_2, \dots, F_N\}$) 需要进行覆盖测试，每个特性都被评估了对应的优先级，特性使用其 ID 作为下标进行标识。

设计了 M 个测试用例 ($\{T_1, T_2, \dots, T_M\}$)，每个用例对应了一个覆盖特性的集合，测试用例使用其 ID 作为下标进行标识，测试用例的优先级定义为其覆盖的特性的优先级之和。

在开展测试之前，需要制定测试用例的执行顺序，规则为：优先级大的用例先执行，如果存在优先级相同的用例，用例 ID 小的先执行。

输入描述：

第一行输入为 N 和 M ， N 表示特性的数量， M 表示测试用例的数量，

$0 < N \leq 100, 0 < M \leq 100$ 。

之后 N 行表示特性 ID=1 到特性 ID= N 的优先级。

再接下来 M 行表示测试用例 ID=1 到测试用例 ID= M 关联的特性的 ID 的列表。

输出描述：

按照执行顺序（优先级从大到小）输出测试用例的 ID，每行一个 ID。

补充说明：

测试用例覆盖的 ID 不重复。

。

示例1

输入: 5 4

1
1
2
3
5
1 2 3
1 4
3 4 5
2 3 4

输出: 3

4
1
2

说明: 测试用例的优先级计算如下:

$$T_1 = P_{F1} + P_{F2} + P_{F3} = 1 + 1 + 2 = 4$$

$$T_2 = P_{F1} + P_{F4} = 1 + 3 = 4$$

$$T_3 = P_{F3} + P_{F4} + P_{F5} = 2 + 3 + 5 = 10$$

$$T_4 = P_{F2} + P_{F3} + P_{F4} = 1 + 2 + 3 = 6$$

按照优先级从小到大, 以及相同优先级, ID小的先执行的规则, 执行顺序为T3,T4,T1,T2

示例2

输入: 3 3

3
1
5
1 2 3
1 2 3
1 2 3

输出: 1

2
3

说明: 测试用例的优先级计算如下:

$$T_1 = P_{F1} + P_{F2} + P_{F3} = 3 + 1 + 5 = 9$$

$$T_2 = P_{F1} + P_{F2} + P_{F3} = 3 + 1 + 5 = 9$$

$$T_3 = P_{F1} + P_{F2} + P_{F3} = 3 + 1 + 5 = 9$$

每个优先级一样, 按照ID从小到大执行, 执行顺序为T1,T2,T3

```
import java.util.Arrays;  
import java.util.Scanner;
```

// 注意类名必须为 Main, 不要有任何 package xxx 信息

```
public class Main {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        // 注意 hasNext 和 hasNextLine 的区别  
        int n = in.nextInt();  
        int m = in.nextInt();  
        int[] priorities = new int[n];  
        for (int i = 0; i < n; i++) {  
            priorities[i] = in.nextInt();  
        }  
        in.nextLine();  
        int[][] res = new int[m][2];  
        for (int i = 0; i < m; i++) {  
            int sum = Arrays.stream(in.nextLine().split(" ")).mapToInt(a -> {
```

```

        int idx = Integer.parseInt(a);
        return priorities[idx - 1];
    }).sum();
    res[i][0] = sum;
    res[i][1] = i;
}
Arrays.sort(res, (a,b) -> {
    return b[0]-a[0] == 0 ? a[1] - b[1] : b[0]-a[0];
});
for (int[] re : res) {
    System.out.println(re[1]+1);
}
}
}

```