

排队游戏题目描述：

新来的老师给班里的同学排一个队。每个学生有一个能力值。一些学生是刺头，不会听老师的话，自己选位置；非刺头同学在剩下的位置按照能力值从小到大排。

对于非刺头同学，如果发现他前面有能力值比自己高的同学，他不满程度就增加，增加的数量等于前面能力值比他大的同学的个数。刺头不会产生不满。

如果整个班级累计的不满程度超过 k ，那么老师就没有办法教这个班级了。

输入描述：

输入有三行：

第一行为 n, m, k ，空格隔开，分别表示班级总人数，刺头人数，最大不满程度 k 。

第二行为刺头所在位置（从 0 开始，即排队数组的下标，比如 1 代表队伍中第 2 个同学是刺头），位置的数组也是排序的。

第三行有 n 个数，空格隔开，表示老师排好的队中每个人的能力值，其中非刺头同学一定按照能力值从小到大排好序的。

输出描述：

0 表示老师可以继续教这个班级

1 表示老师无法继续教这个班级

补充说明：

n 范围是 $[1, 100000]$

m 范围是 $[1, n]$

k 范围是 $[1, 1000000000]$

每位同学的能力值范围是 $[1000, 100000]$

```
#include <iostream>
```

```
#include <string>
```

```
#include <map>

#include <vector>

#include <queue>

#include <cmath>

#include <algorithm>

#include <utility>

#include <random>
```

```
#define DEBUG
```

```
int main() {

    int N;

    int c_n;

    int k;

    std::cin >> N >> c_n >> k;


    std::vector<int> c_flags(N, 0);

    for (int i = 0; i < c_n; i++) {

        int n;

        std::cin >> n;

        c_flags[n] = 1;

    }
```

```
std::vector<int> scores;

for (int i = 0; i < N; i++) {

    int n;

    std::cin >> n;

    scores.push_back(n);

}
```

```
std::vector<int> k_list(N, 0);

std::priority_queue<int, std::vector<int>, std::greater<int>> queue;

for (int i = 0; i < N; i++) {

    if (c_flags[i] == 1) {

        queue.push(scores[i]);

    } else {

        if (i == 0) {

            k_list[i] = 0;

            continue;

        }


```

```
        while (!queue.empty()) {

            int max_score = queue.top();

            if (max_score <= scores[i]) {
```

```

        queue.pop();

    } else {

        break;

    }

}

k_list[i] = queue.size();

}

}

int total = 0;

for (int i = 0; i < N; i++) {

    total += k_list[i];

}

std::cout << (total > k ? 1 : 0) << std::endl;

return 0;

}

```