Problem F
Professor and His Overworked Minions II

Time limit: **3 seconds**
Memory limit: **1024 megabytes**

Problem Description

Professor Lee hopes to provide his students with more learning opportunities, so he plans to assign some of the laboratory's tasks to them. However, the students are already busy with their coursework and jobs, and the number of tasks they can take on is limited.

To ensure the students' quality of life and mental well-being, Professor Lee sets the maximum busyness level for each student at 10. For example, if Allen's busyness level is 8, it means she can take on at most 2 points of workload (such as two 1-point tasks or one 2-point task). If this limit is exceeded, Allen will immediately start playing screaming marmot videos, which will cause everyone in the lab to be unable to work.

To distribute the work fairly, Professor Lee limits each student to at most one task. He also hopes to make full use of everyone's available time. Please help him determine how many students in the lab will have their workload levels exactly reach the maximum after all tasks are assigned.

Input Format

Your program is to read from standard input. The input may contain multiple test cases. Each test case is given in three lines. The first line contains two positive integers N and M, representing the number of students and the number of tasks. The second line contains N integers $S_i (1 \leq i \leq N)$, where each value represents the busyness level of a student. The third line contains M integers $T_j (1 \leq j \leq M)$, where each value represents the workload level of a task. The input ends when N is '`0`'.

Output Format

Your program should write to standard output. For each test case, output one line containing an integer that represents the number of students whose workload levels reach the maximum after the tasks are assigned.

Technical Specification

- $0 \leq N \leq 20$
- $1 \leq M \leq 20$
- $0 \leq S_i \leq 10, 1 \leq i \leq N$

- $1 \leq T_j \leq 10, 1 \leq j \leq M$

Sample Input 1

```
3 6
2 2 10
5 8 9 6 3 9
8 5
7 8 8 2 6 2 5 8
4 2 10 1 8
0
```

Sample Output 1

```
2
3
```