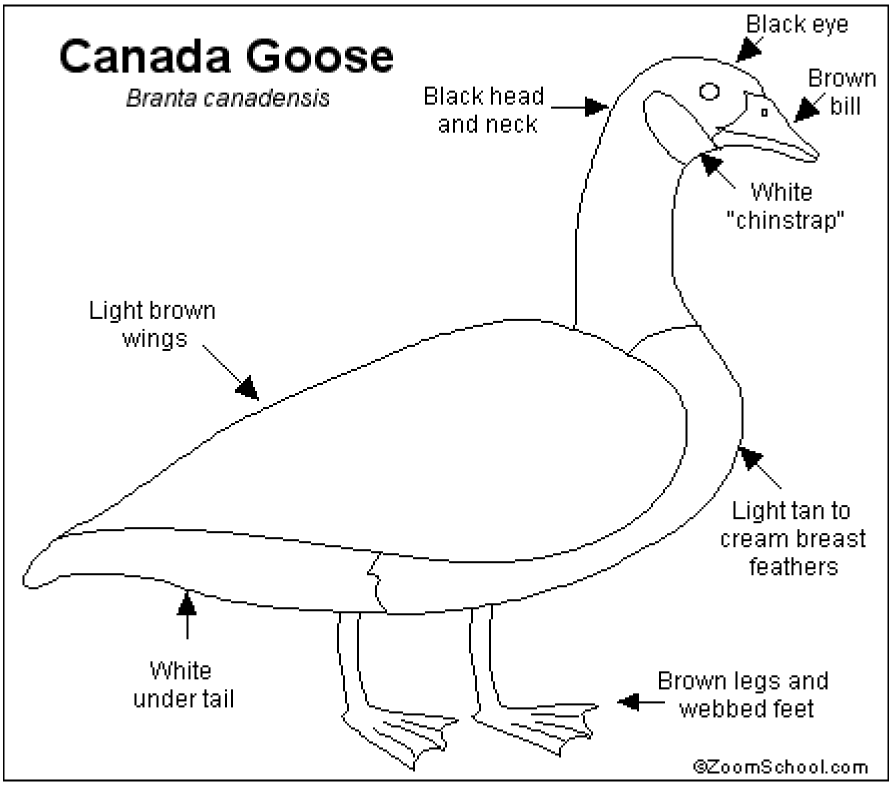
Problem C: The Dragon of Loowater



Once upon a time, in the Kingdom of Loowater, a minor nuisance turned into a major problem.

The shores of Rellau Creek in central Loowater had always been a prime breeding ground for geese. Due to the lack of predators, the geese population was out of control. The people of Loowater mostly kept clear of the geese. Occasionally, a goose would attack one of the people, and perhaps bite off a finger or two, but in general, the people tolerated the geese as a minor nuisance.

One day, a freak mutation occurred, and one of the geese spawned a multi-headed fire-breathing dragon. When the dragon grew up, he threatened to burn the Kingdom of Loowater to a crisp. Loowater had a major problem. The king was alarmed, and called on his knights to slay the dragon and save the kingdom.

The knights explained: "To slay the dragon, we must chop off all its heads. Each knight can chop off one of the dragon's heads. The heads of the dragon are of different sizes. In order to chop off a head, a knight must be at least as tall as the diameter of the head. The knights' union demands that for chopping off a head, a knight must be paid a wage equal to one gold coin for each centimetre of the knight's height."

Would there be enough knights to defeat the dragon? The king called on his advisors to help him decide how many and which knights to hire. After having lost a lot of money building Mir Park, the king wanted to minimize the expense of slaying the dragon. As one of the advisors, your job was to help the king. You took it very seriously: if you failed, you and the whole kingdom would be burnt to a crisp!

Input Specification:

The input contains several test cases. The first line of each test case contains two integers between 1 and 20000 inclusive, indicating the number *n* of heads that the dragon has, and the number *m* of knights in the kingdom. The next *n* lines each contain an integer, and give the diameters of the dragon's heads, in centimetres. The following *m* lines each contain an integer, and specify the heights of the knights of Loowater, also in centimetres.

The last test case is followed by a line containing:

```
0 0
```

Output Specification:

For each test case, output a line containing the minimum number of gold coins that the king needs to pay to slay the dragon. If it is not possible for the knights of

Loowater to slay the dragon, output the line:

```
Loowater is doomed!
```

Sample Input:

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

Output for Sample Input:

```
11
Loowater is doomed!
```

Ondřej Lhoták

Problem F
"Advertisement"

The Department of Recreation has decided that it must be more profitable, and it wants to sell advertising space along a popular jogging path at a local park. They have built a number of billboards (special signs for advertisements) along the path and have decided to sell advertising space on these billboards. Billboards are situated evenly along the jogging path, and they are given consecutive integer numbers corresponding to their order along the path. At most one advertisement can be placed on each billboard.

A particular client wishes to purchase advertising space on these billboards but needs guarantees that every jogger will see it's advertisement at least K times while running along the path. However, different joggers run along different parts of the path.

Interviews with joggers revealed that each of them has chosen a section of the path which he/she likes to run along every day. Since advertisers care only about billboards seen by joggers, each jogger's personal path can be identified by the sequence of billboards viewed during a run. Taking into account that billboards are numbered consecutively, it is sufficient to record the first and the last billboard numbers seen by each jogger.

Unfortunately, interviews with joggers also showed that some joggers don't run far enough to see K billboards. Some of them are in such bad shape that they get to see only one billboard (here, the first and last billboard numbers for their path will be identical). Since out-of-shape joggers won't get to see K billboards, the client requires that they see an advertisement on every billboard along their section of the path. Although this is not as good as them seeing K advertisements, this is the best that can be done and it's enough to satisfy the client.

In order to reduce advertising costs, the client hires you to figure out how to minimize the number of billboards they need to pay for and, at the same time, satisfy stated requirements.

Input

The first line of the input consist of an integer indicating the number of test cases in theinput. Then there's a blank line and the test cases separated by a blank line.

The first line of each test case contains two integers K and N ($1 \leq K, N \leq 1000$) separated by a space. K is the minimal number of advertisements that every jogger must see, and N is the total number of joggers.

The following N lines describe the path of each jogger. Each line contains two integers A_i and B_i (both numbers are not greater than 10000 by absolute value). A_i represents the first billboard number seen by jogger number i and B_i gives the last billboard number seen by that jogger. During a run, jogger i will see billboards A_i, B_i and all billboards between them.

Output

On the first line of the output fof each test case, write a single integer M. This number gives the minimal number of advertisements that should be placed on billboards in order to fulfill the client's requirements. Then write M lines with one number on each line. These numbers give (in ascending order) the billboard numbers on which the client's advertisements should be placed. Print a blank line between test cases.

Sample input

```
1

5 10
1 10
20 27
0 -3
15 15
8 2
7 30
-1 -10
27 20
2 9
14 21
```

Sample output for the sample input

```
19
-5
-4
-3
-2
-1
0
4
5
6
7
8
15
18
19
20
21
25
26
27
```

Problem D
The Grand Dinner

Input: standard input
Output: standard output
Time Limit: 15 seconds
Memory Limit: 32 MB

Each team participating in this year's **ACM World Finals** contest is expected to join the grand dinner to be arranged after the prize giving ceremony ends. In order to maximize the interaction among the members of different teams, it is expected that no two members of the same team sit at the same table.

Now, given the number of members in each team (including contestants, coaches, reserves, guests etc.) and the seating capacity of each available table, you are to determine whether it is possible for the teams to sit as described in the previous paragraph. If such an arrangement is possible you must also output one possible seating arrangement. If there are multiple possible arrangements, any one is acceptable.

Input

The input file may contain multiple test cases. The first line of each test case contains two integers **M** $(1 \le M \le 70)$ and **N** $(1 \le N \le 50)$ denoting the number of teams and the number of tables respectively. The second line of the test case contains **M** integers where the **i**-th $(1 \le i \le M)$ integer m_i $(1 \le m_i \le 100)$ indicates the number of members of team **i**. The third line contains **N** integers where the **j**-th $(1 \le j \le N)$ integer n_j $(2 \le n_j \le 100)$ indicates the seating capacity of table **j**.

A test case containing two zeros for **M** and **N** terminates the input.

Output

For each test case in the input print a line containing either **1** or **0** depending on whether or not there exists a valid seating arrangement of the team members. In case of a successful arrangement print **M** additional lines where the **i**-th $(1 \le i \le N)$ of these lines contains a table number (an integer from **1** to **N**) for each of the members of team **i**.

Sample Input

```
4 5
4 5 3 5
3 5 2 6 4
4 5
4 5 3 5
3 5 2 6 3
0 0
```

Sample Output

```
1
1 2 4 5
1 2 3 4 5
2 4 5
1 2 3 4 5
0
```

(World Finals Warm-up Contest, Problem Setter: Rezaul Alam Chowdhury)

Problem C : Fill the Containers

Time Limit: 1 sec

A conveyor belt has a number of vessels of different capacities each filled to brim with milk. The milk from conveyor belt is to be filled into 'm' containers. The constraints are:

- Whenever milk from a vessel is poured into a container, the milk in the vessel must be completely poured into that container only. That is milk from same vessel can not be poured into different containers.
- The milk from the vessel must be poured into the container in order which they appear in the conveyor belt. That is, you cannot randomly pick up a vessel from the conveyor belt and fill the container.
- The ith container must be filled with milk only from those vessels that appear earlier to those that fill jth container, for all i < j

Given the number of containers 'm', you have to fill the containers with milk from all the vessels, without leaving any milk in the vessel. The containers need not necessarily have same capacity. You are given the liberty to assign any possible capacities to them. Your job is to find out the minimal possible capacity of the container which has maximal capacity. (If this sounds confusing, read down for more explanations.)

Input Format

A single test case consist of 2 lines. The first line specifies 1<=n<=1000 the number of vessels in the conveyor belt and then 'm' which specifies the number of containers to which, you have to transfer the milk. (1 <= m <= 1000000) .The next line contains, the capacity 1<=c<=1000000 of each vessel in order which they appear in the conveyor belt. Note that, milk is filled to the brim of any vessel. So the capacity of the vessel is equal to the amount of milk in it. There are several test cases terminated by EOF.

Output Format

For each test case, print the minimal possible capacity of the container with maximal capacity. That is there exists a maximal capacity of the containers, below which you can not fill the containers without increasing the number of containers. You have to find such capacity and print it on a single line.

Sample Input

5 3
1 2 3 4 5
3 2
4 78 9

Sample Output

6
82

Explanation of the output:

Here you are given 3 vessels at your disposal, for which you are free to assign the capacity. You can transfer, {1 2 3} to the first container, {4} to the second, {5} to third. Here the maximal capacity of the container is the first one which has a capacity of 6. Note that this is optimal too. That is, you can not have the maximal container, have a capacity, less than 6 and still use 3 containers only and fill the containers with all milk.

For the second one, the optimal way is, {4 78} into the first container, and {9} to the second container. So the minimal value of the maximal capacity is 82. Note that {4} to first container and {78 9} to the second is not optimal as, there exists a way to have an assignement of maximal capacity to 82, as opposed to 87 in this case.

Problem Setter: Rajesh S R
Written for CarteBlanche '08

Problem E: Marbles on a tree

n boxes are placed on the vertices of a rooted tree, which are numbered from 1 to *n*, $1 \le n \le 10000$. Each box is either empty or contains a number of marbles; the total number of marbles is *n*.

The task is to move the marbles such that each box contains exactly one marble. This is to be accomplished be a sequence of moves; each move consists of moving one marble to a box at an adjacent vertex. What is the minimum number of moves required to achieve the goal?

The input contains a number of cases. Each case starts with the number *n* followed by *n* lines. Each line contains at least three numbers which are: *v* the number of a vertex, followed by the number of marbles originally placed at vertex *v* followed by a number *d* which is the number of children of *v*, followed by *d* numbers giving the identities of the children of *v*.

The input is terminated by a case where $n = 0$ and this case should not be processed.

For each case in the input, output the smallest number of moves of marbles resulting in one marble at each vertex of the tree.

Sample input

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

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

Output for sample input

```
7
14
20
```

P. Rudnicki