Problem A
Team Work
Number of test cases: 25
Time limit: 1 second

You are the newly appointed coach of programming club. The club has n talented students, each has an unique integer skill rating based on their performance in previous contests. No two students have the same skill rating. For the upcoming National Collegiate Programming Contest (NCPC), you need to select exactly m ($\leq n$) students to form a team. However, teams with large skill gaps tend to have coordination issues. To ensure fair competition and promote teamwork, you decide to select the team so that the difference between the highest and lowest skill ratings of the team member must be as small as possible.

Given the skill ratings of all n students and the requirement to select m team members, determine the minimum possible skill gap (difference between the highest and lowest ratings) you can achieve in your team selection.

Input Format

Input consists of multiple test cases. For each test case: First line contains two integers n and m, where $3 \leq n \leq 30000$ is the total number of students and $3 \leq m \leq 10000$ is the required team size. Second line contains n distinct space-separated integers representing each student's skill rating.

The last line of the input contains only two zeros (`0 0`) to indicate the end of the input.

Output Format

For each test case, print on a line a single integer representing the minimum possible skill gap for the optimal team selection.

Sample Input

```
5 3
450 1200 1800 1500 2400
6 4
240 360 60 120 600 900
0 0
```

Output for the Sample Input

```
600
300
```

Problem B
Festival Route Planning
Number of test cases: 10
Time limit: 1 second

The annual Spring Festival is approaching in the city of Algorithmica! The city is laid out as an $n \times m$ grid, where each cell represents a district. As the festival coordinator, you need to plan the optimal route from the city entrance at the top-left corner $(1, 1)$ to the festival grounds at the bottom-right corner (n, m).

Each district (i, j) offers a unique festival experience with a *happiness value* $h_{i,j}$ that visitors gain when passing through it. The happiness value can be positive (enjoyable attractions), negative (construction zones), or zero (neutral areas).

Due to the parade routes and one-way festival decorations, visitors can only move in two directions:

- **Right**: from district (i, j) to $(i, j + 1)$
- **Down**: from district (i, j) to $(i + 1, j)$

Additionally, the festival safety committee has assigned each district a *crowd level* $c_{i,j}$ $(1 \leq c_{i,j} \leq 3)$, where:

- Level 1: Low crowd density
- Level 2: Moderate crowd density
- Level 3: High crowd density

To ensure visitor safety and comfort, the committee has established a strict rule: **visitors cannot pass through two consecutive districts that both have crowd level 3**.

Your task is to find a valid path from the entrance to the festival grounds that maximizes the total happiness gained along the route.

Input Format

Input consists of multiple test cases. For each test case:

- The first line contains two integers n and m $(2 \leq n, m \leq 1000)$, the dimensions of the city grid.
- The next n lines each contain m integers. The ith line jth integer $h_{i,j}$ $(-1000 \leq h_{i,j} \leq 1000)$ represents the happiness value of district (i, j).

- The next n lines each contain m integers. The ith line jth integer $c_{i,j}$ ($1 \le c_{i,j} \le 3$) represents the crowd level of district (i, j).

The last line of the input contains only two zeros (`0 0`) to indicate the end of the input.

Output Format

For each test case, print a single integer: the maximum total happiness that can be achieved on any valid path from $(1, 1)$ to (n, m). If no valid path exists, print `IMPOSSIBLE`.

Sample Input

```
3 4
1 3 2 5
2 -1 4 1
3 2 2 3
1 3 2 3
2 1 3 1
3 2 1 2
2 3
10 -5 8
-2 3 6
3 3 3
3 1 3
0 0
```

Output for the Sample Input

```
16
IMPOSSIBLE
```

Explanation of the Above Sample Input Cases

Sample Input 1:

- Grid dimensions: 3×4
- The optimal path is: $(1, 1) \to (1, 2) \to (1, 3) \to (1, 4) \to (2, 4) \to (3, 4)$. Crowd levels on this path are: $1 \to 3 \to 2 \to 3 \to 1 \to 2$ (valid: no consecutive 3s). And the Happiness collected are $1 + 3 + 2 + 5 + 1 + 3 = 16$.

Sample Input 2:

- Grid dimensions: 2×3
- Note that all districts except $(2, 2)$ have crowd level 3. Any path from $(1, 1)$ to $(2, 3)$ will pass through consecutive districts with crowd level 3. Therefore, no valid path exists.

Problem C
Minimum Number of Classrooms
Number of test cases: 5
Time limit: 1 second

You are an administrative staff member responsible for scheduling and classroom assignments in a university. There are a number of activities scheduled throughout the day. Each activity has a fixed start and end time. Every activity must be assigned to one classroom, and a classroom can only host one activity at a time.

Due to space constraints, you would like to use as few classrooms as possible, while still ensuring that no two overlapping activities are assigned to the same room.

Given the schedule of all activities, your task is to determine the minimum number of classrooms required to host all activities without conflicts.

Each activity is represented by a time interval $[s_i, e_i)$, meaning it starts at time s_i and ends at e_i. The end time is exclusive, meaning the room becomes available again at e_i.

Write an efficient program that computes the minimum number of classrooms needed to hold all activities without overlap.

Input File Format

Input consists of multiple test cases. For each test case, the first line contains an integer n $(1 \leq n \leq 10^5)$ — the number of activities. Each of the next n lines contains two integers s_i and e_i $(0 \leq s_i < e_i \leq 10^9)$ — the start and end times of an activity. The last line of the input contains only a zero (`0`) to indicate the end of the input.

Output Format

For each test case, print on a single line a single integer — the minimum number of classrooms required.

Sample Input

```
5
1 4
2 5
5 8
6 9
```

```
8 10
0
```

Output for the Sample Input

```
2
```

Problem D
Maintenance
Number of test cases: 9
Time limit: 3 seconds

National Center for Programming and Computation (NCPC) has n departments, numbered from 1 to n, and each department has it own copy machine, rented from the Yeroy company. Every October, the company sends m engineers, numbered from 1 to m, to NCPC for the copy machine maintenance. The engineers have the same ability. That is, for an identical machine, any two engineers use the same time for maintenance. The departments may rent different models, so the required maintenance times may also vary. It is known that the machine in department i needs time t_i for maintenance, and there are k different models in NCPC. The manager of Yeroy wants to assign the n machines to the m engineers so that they can finish their job as early as possible. Please write a program to find the minimum possible time they have to stay in NCPC. Precisely, let E be the set of engineers, and let M_j be the set of machines assigned to engineer j, you are asked to find

$$\max_{j \in E} \sum_{i \in M_j} t_i.$$

Input File Format

Input consists of multiple test cases. For each test case, the first line contains two positive integers n and m, which are the numbers of machines and engineers, respectively. The second line contains n positive integers $t_1, \ldots, t_n$, which are the required times for maintenance. ($5 \leq n \leq 50$, $2 \leq m \leq 10$, $1 \leq t_i \leq 1000$ for $1 \leq i \leq n$)

The last line of the input contains only two zeros (0 0) to indicate the end of the input.

Output Format

For each test case, print on a line a single positive integer, which is the minimum possible time the engineers have to stay in NCPC.

Sample Input

```
7 3
3 1 4 1 5 3 3
0 0
```

Output for the Sample Input

```
7
```

Problem E
Trip Scheduling
Number of test cases: ≤ 20
Time limit: 1 second

Tom is scheduling m trips over n days. The trips are labelled from 0 to $m-1$. For $0 \leq i < m$, the i-th trip needs a period of t_i consecutive days. The trips should be scheduled in the given order and can not overlap each other. Furthermore, there should be a break of at least q days between any two trips to rest up. That is, if the i-th trip starts at day s_i, then it will ends at day $s_i + t_i - 1$, and the starting time of the $i+1$-th trip must satisfy $s_{i+1} \geq s_i + t_i + q$. Tom has evaluated the comfort level of each of the next n days for taking trips. For $0 \leq i < n$, p_i is the comfort level of i-th day, and he wants to schedule the trips to maximize the minimal comfort level of the days in his trips.

The following is an example of comfort level data for $n = 10$.

day	0	1	2	3	4	5	6	7	8	9
p_i	2	7	5	4	3	3	9	6	7	4

Suppose that Tom has two trips with $t_0 = 2$ and $t_1 = 3$, and the required minimal gap $q = 2$. Then, the best scheduling is days $[1, 2]$ for trip 0 and $[6, 8]$ for trip 1. The maximized minimal comfort level is $\min\{7, 5, 9, 6, 7\} = 5$.

As another example, suppose that $q = 5$ while $t_0 = 2$, $t_1 = 3$, and the comfort levels are the same. The best scheduling is $[0, 1]$ for trip 0 and $[7, 9]$ for trip 1. The maximized minimal comfort level is 2, and the scheduling is also the only valid one since $t_0 + q + t_1 = 10 = n$.

Write an efficient program for Tom to compute the maximized minimal comfort level.

Input Format

There are more than one test cases in the input. Each test case starts with three integers m, n and q, where $1 \leq m \leq 10^4$, $1 \leq n \leq 10^5$ and $0 \leq q \leq 10^4$. The second line of each test case contains m positive integers $t_0, t_1, \ldots t_{m-1}$, and the third line contains n integers $p_0, p_1, \ldots, p_{n-1}$, where $0 \leq p_i \leq 10^8$. Two consecutive numbers in the same line are separated by one space. For each test case, $\sum t_i + (n-1)q \leq n$, i.e., a valid scheduling always exists.

The total number of days in all test case is at most 1.2×10^6. The last line of the input contains three 0s to indicate the end of the input.

Output Format

For each test case, print the maximized minimal comfort level in one line.

Sample Input

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

Output for the Sample Input

```
5
2
3
```

Problem F
Network Capacity and Pipe Constraints
Number of test cases: 10
Time limit: 5 seconds

In the bustling city of Aquapolis, fresh water flows from a grand central water tower to every neighborhood in town. The city's plumbing system is a masterpiece of civil engineering: from the central water tower, water travels through a web of pipes, branching out so that each neighborhood receives its supply through exactly one unique route from the tower.

Each pipe connects two neighborhoods: water flows from A to B, with a specific capacity C—the maximum amount of water it can safely deliver. Over time, as the city expands and infrastructure is upgraded, the water technicians may change the capacity of any existing pipe.

As part of the mayor's mission to keep Aquapolis hydrated, you are regularly asked to help answer a crucial question: A district is composed of a neighborhood and all the neighborhoods that can be reached downstream by following the pipes. What is the weakest link of a given district? In other words, what is the smallest capacity pipe that would flow water from the central water tower to any neighborhood in that district?

The mayor also wants you to efficiently process reports when workers upgrade or downgrade the capacity of a specific pipe.

Write a program that, given the initial layout of Aquapolis's water system and a sequence of events (either a pipe's capacity being updated, or a question from the mayor about a district's weakest link), reports the answer for each of the mayor's questions.

Input File Format

A test file contains multiple test cases.

For each test case: The first line contains a single integer n ($1 \leq n \leq 10^5$), the number of neighborhoods.

The next $n-1$ lines each contain three integers A, B, and C ($1 \leq A, B \leq n$, $A \neq B$, $1 \leq C \leq 10^9$), describing a directed pipe of capacity C that carries water from neighborhood A to neighborhood B.

The water system forms a tree directed away from Neighborhood 1 (the central water tower).

The next line contains an integer m ($1 \leq m \leq 10^5$), the number of events.

Each of the following m lines describes an event in one of two forms:

- `1 X Y C`: The water technicians have just set a new capacity C for the pipe from X to Y.
- `2 D`: The mayor wants to know: for the district led by neighborhood D, what is the smallest capacity pipe on any route from the central water tower (Neighborhood 1) to any home in that district?

It is guaranteed that in every update query, the pipe from X to Y exists. The last line of the input file contains only an integer 0 to indicate the end of the input.

Output Format

For each mayoral query (`2 D`), print a single integer: the smallest capacity pipe on any route from the central water tower (Neighborhood 1) to any home in the district led by D. Print each answer on its own line.

Sample Input

```
5
1 2 4
2 3 2
2 4 3
1 5 6
5
2 2
1 2 4 1
2 1
2 2
2 5
0
```

Output for the Sample Input

```
2
1
1
6
```