Coder-Strike 2014 - Round 2

A. Data Recovery

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Not so long ago company R2 bought company R1 and consequently, all its developments in the field of multicore processors. Now the R2 laboratory is testing one of the R1 processors.

The testing goes in n steps, at each step the processor gets some instructions, and then its temperature is measured. The head engineer in R2 is keeping a report record on the work of the processor: he writes down the minimum and the maximum measured temperature in his notebook. His assistant had to write down all temperatures into his notebook, but (for unknown reasons) he recorded only m.

The next day, the engineer's assistant filed in a report with all the m temperatures. However, the chief engineer doubts that the assistant wrote down everything correctly (naturally, the chief engineer doesn't doubt his notes). So he asked you to help him. Given numbers n, m, min, max and the list of m temperatures determine whether you can upgrade the set of m temperatures to the set of n temperatures (that is add n - m temperatures), so that the minimum temperature was min and the maximum one was max.

Input

The first line contains four integers n, m, min, max $(1 \le m < n \le 100; 1 \le min < max \le 100)$. The second line contains m space-separated integers t_i $(1 \le t_i \le 100)$ — the temperatures reported by the assistant.

Note, that the reported temperatures, and the temperatures you want to add can contain equal temperatures.

Output

If the data is consistent, print '`Correct`' (without the quotes). Otherwise, print '`Incorrect`' (without the quotes).

Sample test(s)

input
2 1 1 2 1
output
Correct

input
3 1 1 3 2
output
Correct

input
2 1 1 3 2
output
Incorrect

Note

In the first test sample one of the possible initial configurations of temperatures is [1, 2].

In the second test sample one of the possible initial configurations of temperatures is [2, 1, 3].

In the third test sample it is impossible to add one temperature to obtain the minimum equal to 1 and the maximum equal to 3.

B. Spyke Chatting

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The R2 company has n employees working for it. The work involves constant exchange of ideas, sharing the stories of success and upcoming challenging. For that, R2 uses a famous instant messaging program Spyke.

R2 has m Spyke chats just to discuss all sorts of issues. In each chat, some group of employees exchanges messages daily. An employee can simultaneously talk in multiple chats. If some employee is in the k-th chat, he can write messages to this chat and receive notifications about messages from this chat. If an employee writes a message in the chat, all other participants of the chat receive a message notification.

The R2 company is conducting an audit. Now the specialists study effective communication between the employees. For this purpose, they have a chat log and the description of chat structure. You, as one of audit specialists, are commissioned to write a program that will use this data to determine the total number of message notifications received by each employee.

Input

The first line contains three space-separated integers n, m and k $(2 \le n \le 2 \cdot 10^4; 1 \le m \le 10; 1 \le k \le 2 \cdot 10^5)$ — the number of the employees, the number of chats and the number of events in the log, correspondingly.

Next n lines contain matrix a of size $n \times m$, consisting of numbers zero and one. The element of this matrix, recorded in the j-th column of the i-th line, (let's denote it as a_{ij}) equals 1, if the i-th employee is the participant of the j-th chat, otherwise the element equals 0. Assume that the employees are numbered from 1 to n and the chats are numbered from 1 to m.

Next k lines contain the description of the log events. The i-th line contains two space-separated integers x_i and y_i $(1 \le x_i \le n; 1 \le y_i \le m)$ which mean that the employee number x_i sent one message to chat number y_i. It is guaranteed that employee number x_i is a participant of chat y_i. It is guaranteed that each chat contains at least two employees.

Output

Print in the single line n space-separated integers, where the i-th integer shows the number of message notifications the i-th employee receives.

Sample test(s)

input

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

output

```
3 3 1
```

input

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

output

```
0 2 3 0
```

C. Jeopardy!

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

'Jeopardy!' is an intellectual game where players answer questions and earn points. Company Q conducts a simplified 'Jeopardy!' tournament among the best IT companies. By a lucky coincidence, the old rivals made it to the finals: company R1 and company R2.

The finals will have n questions, m of them are auction questions and $n - m$ of them are regular questions. Each question has a price. The price of the i-th question is a_i points. During the game the players chose the questions. At that, if the question is an auction, then the player who chose it can change the price if the number of his current points is strictly larger than the price of the question. The new price of the question cannot be less than the original price and cannot be greater than the current number of points of the player who chose the question. The correct answer brings the player the points equal to the price of the question. The wrong answer to the question reduces the number of the player's points by the value of the question price.

The game will go as follows. First, the R2 company selects a question, then the questions are chosen by the one who answered the previous question correctly. If no one answered the question, then the person who chose last chooses again.

All R2 employees support their team. They want to calculate what maximum possible number of points the R2 team can get if luck is on their side during the whole game (they will always be the first to correctly answer questions). Perhaps you are not going to be surprised, but this problem was again entrusted for you to solve.

Input

The first line contains two space-separated integers n and m $(1 \le n, m \le 100;\ m \le min(n, 30))$ — the total number of questions and the number of auction questions, correspondingly. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^7)$ — the prices of the questions. The third line contains m distinct integers b_i $(1 \le b_i \le n)$ — the numbers of auction questions. Assume that the questions are numbered from 1 to n.

Output

In the single line, print the answer to the problem — the maximum points the R2 company can get if it plays optimally well. It is guaranteed that the answer fits into the integer 64-bit signed type.

Sample test(s)

input
4 1 1 3 7 5 3
output
18

input
3 2 10 3 8 2 3
output
40

input
2 2 100 200 1 2
output
400

D. 2048

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The programmers from the R2 company love playing 2048. One day, they decided to invent their own simplified version of this game — 2^k on a stripe.

Imagine an infinite in one direction stripe, consisting of unit squares (the side of each square is equal to the height of the stripe). Each square can either be empty or contain some number.

Initially, all squares are empty. Then at infinity one of the unit squares number 2 or 4 appears. Then the player presses a button once, and the appeared number begins to move towards the beginning of the stripe. Let's assume that some number x moves to the beginning of the stripe, then it will stop if:

1. it either gets in the first square of the stripe;
2. or it is in the square that is preceded by a square with number y $(y \neq x)$. But if number x at some point of time gets to the square with the same number then both numbers add to each other and result in $2x$. The new number $2x$ continues moving to the beginning of the stripe by the same rules.

After the final stop of the number moving process, the infinity gets a new number 2 or 4 and the process repeats. Read the notes to the test samples to better understand the moving strategy.

I guess you've understood that the game progress fully depends on the order in which numbers 2 and 4 appear. Let's look at some sequence of numbers 2 and 4 in the game. We assume that the sequence is *winning* if it results in at least one square getting the number greater or equal than 2^k.

The goal of the game is to make up a winning sequence of n numbers. But not everything is so simple, some numbers in the sequence are identified beforehand. You are given a sequence consisting of numbers 0, 2, 4. Count how many ways there are to replace each 0 of the sequence with 2 or 4 to get a winning sequence.

Input

The first line contains two integers n and k $(1 \leq n \leq 2000; 3 \leq k \leq 11)$. The next line contains sequence of n integers, each of them is either 0, or 2, or 4.

Output

Print a single integer — the number of ways to replace zeroes by numbers 2 or 4 to get a winning sequence. As this number can be rather large, print it modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input
7 4 2 2 4 2 2 2 2
output
1

input
1 3 0
output
0

input
2 3 0 4
output
1

input
5 4 2 0 0 4 4
output
2

Note

Consider the first example. The beginning of the strip will look as follows:

2 → 4 → 8 → 8 2 → 8 4 → 8 4 2 → 16.

To better understand the game, you can see the original game on `http://gabrielecirulli.github.io/2048/`. Please note that the game that is described on the strip is slightly different from the original game (when the two numbers add up in the original game, they do not keep moving). Be careful, the game is addictive, there isn't much time for the contest!

E. Maze 2D

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The last product of the R2 company in the 2D games' field is a new revolutionary algorithm of searching for the shortest path in a $2 \times n$ maze.

Imagine a maze that looks like a $2 \times n$ rectangle, divided into unit squares. Each unit square is either an empty cell or an obstacle. In one unit of time, a person can move from an empty cell of the maze to any side-adjacent empty cell. The shortest path problem is formulated as follows. Given two free maze cells, you need to determine the minimum time required to go from one cell to the other.

Unfortunately, the developed algorithm works well for only one request for finding the shortest path, in practice such requests occur quite often. You, as the chief R2 programmer, are commissioned to optimize the algorithm to find the shortest path. Write a program that will effectively respond to multiple requests to find the shortest path in a $2 \times n$ maze.

Input

The first line contains two integers, n and m $(1 \le n \le 2 \cdot 10^5; 1 \le m \le 2 \cdot 10^5)$ — the width of the maze and the number of queries, correspondingly. Next two lines contain the maze. Each line contains n characters, each character equals either '.' (empty cell), or 'X' (obstacle).

Each of the next m lines contains two integers v_i and u_i $(1 \le v_i, u_i \le 2n)$ — the description of the i-th request. Numbers v_i, u_i mean that you need to print the value of the shortest path from the cell of the maze number v_i to the cell number u_i. We assume that the cells of the first line of the maze are numbered from 1 to n, from left to right, and the cells of the second line are numbered from $n + 1$ to $2n$ from left to right. It is guaranteed that both given cells are empty.

Output

Print m lines. In the i-th line print the answer to the i-th request — either the size of the shortest path or -1, if we can't reach the second cell from the first one.

Sample test(s)

input

```
4 7
.X..
...X
5 1
1 3
7 7
1 4
6 1
4 7
5 7
```

output

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

input

```
10 3
X...X..X..
..X...X..X
11 7
7 18
18 10
```

output

```
9
-1
3
```