

## Problem A. Martian Volleyball

Input file:            `standard input`  
Output file:          `standard output`  
Time limit:           2 seconds  
Memory limit:        256 megabytes

Volleyball match at Mars is played by two teams until one of the teams gets  $k$  points, with score at least 2 points more than the score of the other team. For each ball played exactly one of the teams gets 1 point. Now the score of the first team is  $x$ , the score of the second team is  $y$ . What is the minimal number of balls that must be played until one of the teams wins the match?

### Input

Input contains several test cases. The first line contains integer  $t$  — the number of cases ( $1 \leq t \leq 5000$ ). Each test case is described using one line that contains three integers separated by spaces:  $k$ ,  $x$  and  $y$  ( $1 \leq k \leq 100$ ;  $0 \leq x, y \leq 100$ ). It is guaranteed that the score can be obtained in a correct unfinished game.

### Output

For each test case print one line — the minimal number of balls to be played, until the match is finished.

### Example

| standard input | standard output |
|----------------|-----------------|
| 3              | 1               |
| 2 1 0          | 1               |
| 3 4 3          | 5               |
| 5 0 0          |                 |

## Problem B. Painting the Wall

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          2 seconds  
Memory limit:       256 megabytes

Little girl Masha is looking at the wall in her room. The wall is tiled with square tiles, but some of the tiles are replaced with lamps. So it is possible to consider the wall to be a rectangle of  $n \times m$ , some cells contain tiles, other cells contain lamps.

Masha has paints of  $k$  different colors. Consider continuous segments of tiles, having an edge of the wall or a lamp at each of its ends. Masha wants to paint all tiles in such way, that any such segment has all tiles painted in different colors. Masha will not paint lamps. She doesn't have to use all the colors.

Help Masha to paint the wall.

### Input

Input contains several test cases. The first line contains the number of test cases  $t$ .

Each test case is described by several lines. The first line contains three integers:  $n, m, k$  ( $1 \leq n, m \leq 100$ ,  $1 \leq k \leq \max(n, m)$ ) — the size of the wall and the number of paints Masha has.

The following  $n$  lines contain  $m$  integers  $a_{ij}$  each:

- $a_{ij} = 0$  means that the position  $(i, j)$  contains the lamp;
- $a_{ij} = 1$  means that the position  $(i, j)$  contains the tile.

The total number of tiles and lamps in all test cases of one input doesn't exceed  $10^5$ .

### Output

For each test case first print the answer:

- NO, if there is no way to paint the wall.
- YES, if there is at least one way to paint the wall. In this case the following  $n$  lines must contain  $m$  integers  $b_{ij}$  each — the color of the tile at position  $(i, j)$ , or 0, if there is a lamp at this position. If there are several ways to paint the wall, you can output any one.

### Example

| standard input | standard output |
|----------------|-----------------|
| 2              | YES             |
| 4 3 2          | 0 2 0           |
| 0 1 0          | 2 0 2           |
| 1 0 1          | 1 0 1           |
| 1 0 1          | 0 1 0           |
| 0 1 0          | NO              |
| 3 4 2          |                 |
| 0 1 0 1        |                 |
| 1 0 1 1        |                 |
| 1 1 1 0        |                 |

## Problem C. Magic Artifact

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Maxim is playing a video game. It has  $n$  levels, numbered from 1 to  $n$ . Levels can be completed in any order, it takes Maxim  $a_i$  seconds to complete the  $i$ -th level.

Maxim can find a magic artifact at one of the levels. There is exactly one magic artifact in the game, and once found it will increase the speed of Maxim's hero and reduce the time needed to complete the level. However, it is not known where the artifact is, the probability that it is at the  $i$ -th level is  $p_i$ . The time needed to complete the  $i$ -th level after the artifact is found is  $b_i$  second ( $b_i \leq a_i$ ). Note that artifact doesn't reduce the time needed to complete the level where it is found.

Maxim wants to choose the order he completes the levels, to minimize the expected time to complete the game. Help him to find the minimal possible expected time. Maxim must choose the order to complete the levels before playing the game, the order must not depend on whether the artifact was found or not at some level.

Recall that the expectation of a random variable is the sum over all possible outcomes of a product of the probability of such outcome and the value of the variable. In this problem the outcome corresponds to the level where the artifact is, and the value is the total time needed if the artifact is at that level.

### Input

Input data contains several test cases. The first line contains  $t$  — the number of test cases ( $1 \leq t \leq 1000$ ).

Each test case is described in the following way: the first line contains integer  $n$  — the number of levels ( $1 \leq n \leq 10^5$ ).

The following  $n$  lines describe levels. Each level is specified with three integers  $a_i$ ,  $b_i$  and  $x_i$  — the time to complete the level before the artifact was found, the time to complete it after the artifact was found, and the value that helps to find the probability to find the artifact at that level. The probability is calculated using the formula  $p_i = x_i/10^7$  ( $1 \leq b_i \leq a_i \leq 10^5$ ;  $0 \leq x_i \leq 10^7$ ; the sum of all  $x_i$  is  $10^7$ ).

The sum of values of  $n$  in all test cases of one input data is at most  $5 \cdot 10^5$ .

### Output

For each test case output one floating point value — the expected time to complete the game if the optimal order was chosen. The answer must have an absolute or relative error of at most  $10^{-6}$ .

### Example

| standard input | standard output |
|----------------|-----------------|
| 2              | 16              |
| 3              | 10.25           |
| 10 5 10000000  |                 |
| 5 3 0          |                 |
| 7 3 0          |                 |
| 4              |                 |
| 3 1 2500000    |                 |
| 4 1 2500000    |                 |
| 10 1 2500000   |                 |
| 2 1 2500000    |                 |

## Problem D. Memory Manager

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           **2 seconds**  
Memory limit:        **256 megabytes**

Peter is developing memory manager MEM 2.0 for his offline storage that uses special magnetic 7D blocks. But he has a problem of accessing data in the optimal way.

Peter's memory manager stores  $n$  blocks of data, numbered from 1 to  $n$ , and he has  $q$  queries of accessing one or several of the blocks. Queries must be processed in order they are listed.

To access the data Peter's memory manager has  $k$  pointers, each of them points to some block. Initially Peter can position his pointers at any desired blocks.

MEM 2.0 can immediately access data from any number blocks if each of them is currently has some pointer at it. If it is not the case, the manager must first move the pointers, this operation for the  $i$ -th query takes  $s_i$  milliseconds total to move any number of pointers.

Peter wants to move pointers in such way that the total time of answering all queries was minimal possible. Queries must be processed in order they are listed, changing the order is not allowed. Help him!

Consider sample test cases.

In the first sample test Peter can initially position pointers at blocks 1, 2 and 4 — after that the first two queries are accessed immediately. Before the third query the pointers must be moved to blocks 2, 3 and 5, it takes  $s_3 = 1$  milliseconds, and moving pointers to blocks 1, 3, and 5 before the fourth query takes another  $s_4 = 1$  milliseconds. The total time is  $s_3 + s_4 = 2$  milliseconds.

The second sample test shows that it is sometimes not optimal to make a greedy choice. It is best not to perform two first queries immediately by positioning pointers at 1, 2 and 4 initially, because moving the pointers before the third query would then take 10 milliseconds. The optimal strategy is to first position pointers at blocks 1, 2 and 3, before the second query move them to blocks 1, 3 and 4 in  $s_2 = 1$  milliseconds, and then move them to 1, 3 and 5 in  $s_4 = 3$  milliseconds before the fourth query. The total time is  $s_2 + s_4 = 4$  milliseconds.

### Input

Input data contains several test cases. The first line contains one integer  $t$  — the number of test cases ( $1 \leq t \leq 1000$ ).

Each of the following  $t$  test cases is described in the following way. The first line of the description contains three integers:  $n, k, q$  — the number of blocks, the number of pointers and the number of queries ( $1 \leq k \leq n \leq 10^5, 1 \leq q \leq 10^6$ ).

The following line contains  $q$  integers  $s_i$  — time needed to move pointers if it is performed before the  $i$ -th query ( $1 \leq s_i \leq 10^4$ ).

The following  $q$  lines contain queries in order they must be processed, the  $i$ -th query is described by a line that first contains  $c_i$  — the number of blocks requested, ( $1 \leq c_i \leq k$ ), followed by  $c_i$  integers  $b_{i,j}$  — the numbers of these blocks, given in ascending order ( $1 \leq b_{i,j} \leq n$ ).

It is guaranteed that the sum of all  $n$  of one input data doesn't exceed  $10^5$ , and the sum of all  $c_i$  in all test cases of one input data doesn't exceed  $10^6$ .

### Output

For each test case print one integer — the minimal total time to answer all queries.

## Example

| standard input | standard output |
|----------------|-----------------|
| 2              | 2               |
| 5 3 4          | 4               |
| 1 1 1 1        |                 |
| 1 2            |                 |
| 2 1 4          |                 |
| 2 2 3          |                 |
| 3 1 3 5        |                 |
| 5 3 4          |                 |
| 1 1 10 3       |                 |
| 1 2            |                 |
| 2 1 4          |                 |
| 2 1 3          |                 |
| 3 1 3 5        |                 |

## Problem E. LISA

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          2 seconds  
Memory limit:       256 megabytes

Ilya is working for Laboratory of Investigation of String Algorithms (LISA). Now he is trying to solve the following problem.

You are given an array of strings  $s_1, s_2, \dots, s_n$ , and  $q$  queries. Each query is specified by two integers:  $l_i$  and  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ). To answer the query you must do the following. Let us call a string representable if it is possible to obtain it using the following method: take two strings  $s_x$  and  $s_y$ , where  $l_i \leq x, y \leq r_i$ , take non-empty prefix of  $s_x$  and non-empty suffix of  $s_y$ , concatenate them. The answer to the query is the number of different representable strings for given  $l_i$  and  $r_i$ .

For example, consider  $s = [abc, ab, ac, bcac]$ , take  $l_i = 2, r_i = 3$ . The following strings are representable:

$x = 2, y = 2$ :  $ab = a + b, aab = a + ab, abb = ab + b, abab = ab + ab$ .

$x = 2, y = 3$ :  $ac = a + c, aac = a + ac, abc = ab + c, abac = ab + ac$ .

$x = 3, y = 2$ :  $ab = a + b, aab = a + ab, acb = ac + b, acab = ac + ab$ .

$x = 3, y = 3$ :  $ac = a + c, aac = a + ac, acc = ac + c, acac = ac + ac$ .

So there are 12 different representable strings.

Help Ilya to solve the problem as fast as you can.

### Input

The first line of input contains two integers:  $n$  and  $q$  — the number of strings and the number of queries ( $1 \leq n, q \leq 10^5$ ).

Each of the following  $n$  string contains non-empty words  $s_i$ , that consist of lowercase English letters. The sum of their lengths doesn't exceed  $10^5$ .

The following  $q$  lines contain queries: each line contains two integers  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq n$ ) — parameters of the  $i$ -th query.

### Output

Output  $q$  lines, the  $i$ -th of them must contain one integer — the answer to the  $i$ -th query.

### Example

| standard input | standard output |
|----------------|-----------------|
| 4 3            | 20              |
| abc            | 23              |
| ab             | 12              |
| ac             |                 |
| bcac           |                 |
| 3 4            |                 |
| 1 3            |                 |
| 2 3            |                 |