

Problem A. On the Road to Happiness

Balloon Color: Dark Purple

Mr. AH got his special name from his favorite game, which is called On the Road to Happiness. This game is played with a string s consisting of n uppercase English letters, b beautiful cards, and a magical wand.

A beautiful card ' $x\ y\ c$ ' can be used to swap the letters in positions x and y after paying c coins. Mr. AH can use every beautiful card infinitely, but he must pay each time he wants to use it.

Some of the beautiful cards are interesting, a beautiful card ' $x\ y\ c$ ' is said to be interesting if Mr. AH cannot move the letter in position x to position y (or the letter in position y to position x) using any sequence of cards unless this card is among them.

The magical wand allows Mr. AH to use the beautiful cards without paying anything, except for the interesting cards, Mr. AH must respect these cards and pay for using them. Mr. AH can use the wand as much as he needs.

A position in the given string is said to be happy if it initially contains the letter 'A'. The goal of the game is to find the minimum cost to move all letters 'H' in the string to the happy positions, such that each happy position can contain at most one 'H' letter.

Mr. AH thinks that he is the king of this game, but he wants to see other players play it to prove that. So Mr. AH asked you to play the game, can you?

Input

The first line contains a single integer T ($1 \leq T \leq 100$) specifying the number of test cases.

The first line of each test case contains two integers n and b ($1 \leq n \leq 10^5$, $0 \leq b \leq 2 \times 10^5$), in which n is the length of the string and b is the number of the beautiful cards.

The second line of the each test case contains a string s of length n , consisting of uppercase English letters.

Then b lines follow, each line contains three integers x , y and c ($1 \leq x, y \leq n$, $x \neq y$, $1 \leq c \leq 100$), giving the beautiful cards. It is guaranteed that for each two positions x and y , there is at most one card that can swap the letters in them.

The number of happy positions is between 0 and $\frac{n}{2}$ (inclusive), and it always equals the number of positions that contain the letter 'H'.

It is guaranteed that you can move any letter from its position to any other position.

Output

For each test case, print a single line containing the minimum cost to move all letters 'H' in the string to the happy positions. It is guaranteed that the answer always exists.

Example

Sample Input	Sample Output
2 4 3 HZXA 1 2 5 2 3 2 3 4 3 8 9 HDHBZAZA 1 2 5 1 3 4 2 4 6 3 4 1 4 5 7 5 6 9 5 7 2 6 8 12 7 8 2	10 14

Note

In the second test case, the only interesting beautiful card is '4 5 7', and the letters can be moved by using two sequences of beautiful cards:

- '1 2 5', '2 4 6', '4 5 7', '5 7 2', '7 8 2'
- '1 2 5', '2 4 6', '4 5 7', '5 6 9'

The magic wand will be used on all cards except for the only interesting card, which will be used twice leading to a total cost of 14.

Problem B. Memory Management System

Balloon Color: Light Green

It is your first day in your new job and guesses what, your first task is already waiting for you! Your task is to build a memory management system.

In this system, the memory will consist of m bytes ordered from 1 to m . Originally, there are n files in the system, such that the i^{th} file is occupying all bytes from l_i to r_i (inclusive). It is guaranteed that no two files share the same memory position (byte).

The main goal of the system is to answer multiple queries such that each query consists of a file of size k bytes, and the system must determine if there exist at least k consecutive bytes that can be reserved and assigned to that file.

Formally, at the j^{th} query, you are given an integer k_j and your task is to find a pair of integers l_j and r_j , such that:

- All bytes between l_j and r_j (inclusive) are free (not occupied).
- $r_j - l_j + 1 \geq k_j$.
- If there are multiple solutions, find the one with the maximum r_j . If there are still multiple solutions, find the one with maximum l_j .
- If the j^{th} file cannot be saved in the system, both l_j and r_j must be -1 .

Please note that you are not reserving the bytes for the files in the queries, you are just checking if there exists an empty space for the file in the system. So, an empty byte in the system can be assigned to multiple queries files. However, if a byte is originally occupied in the system, you cannot use it for the queries files.

Input

The first line contains an integer T ($1 \leq T \leq 10$) specifying the number of test cases.

The first line of each test case contains three integers n , m , and q ($0 \leq n \leq m$, $1 \leq m \leq 10^5$, $1 \leq q \leq \min(10^5, m)$), in which n is the number of files in the system originally, m is the number of bytes the system contains, and q is the number of queries.

Then n lines follow, each line contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq m$), giving the files in system. It is guaranteed that no two files share the same memory position (byte).

Then q lines follow, each line contains an integer k_j ($1 \leq k_j \leq m$), giving the queries.

Output

For each query j , print two space-separated integers l_j and r_j , as described in the problem statement. If there are multiple solutions, find the one with the maximum r_j . If there are still multiple solutions, find the one with maximum l_j . If the file cannot be saved in the system, both l_j and r_j must be -1 .

Example

Sample Input	Sample Output
2	2 4
3 9 2	6 7
1 1	4 4
5 5	3 4
8 9	-1 -1
3	
2	
2 5 3	
5 5	
1 2	
1	
2	
4	

Note

In the first test case, the memory system can be represented as “OFFFOFF00”, in which ‘O’ represent an occupied position and ‘F’ represent a free position.

- The first file needs to occupy 3 bytes. So, the only available answer is 2 4.
- The second file needs to occupy 2 bytes. There are 3 available intervals, which are: 2 3, 3 4, and 6 7. The interval that satisfies the problem conditions is 6 7.

Problem C. Large GCD

Balloon Color: Light Orange

This problem is very simple so the problem setters decided that its statement should be simple too. You are given two integers n and m such that $\gcd(n, m) \equiv 1$, and your task is to find the value of the function $F(n, m)$ as follows:

$$F(n, m) = \gcd(5^n + 7^n, 5^m + 7^m)$$

In mathematics, the greatest common divisor (gcd) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers. For example, the gcd of 8 and 12 is 4.

Input

The first line contains an integer T ($1 \leq T \leq 10^5$) specifying the number of test cases.

Each test case consists of a single line contains two integers n and m ($1 \leq n, m \leq 10^9$, $\gcd(n, m) \equiv 1$).

Output

For each test case, print a single line containing the value of the function $F(n, m)$ as described in the statement.

Example

Sample Input	Sample Output
2	2
2 3	12
5 3	

Note

In the first test case, the value of the function can be found as follow:

$$F(2, 3) = \gcd(5^2 + 7^2, 5^3 + 7^3)$$

$$F(2, 3) = \gcd(74, 468)$$

$$F(2, 3) = 2$$

In the second test case, the value of the function can be found as follow:

$$F(5, 3) = \gcd(5^5 + 7^5, 5^3 + 7^3)$$

$$F(5, 3) = \gcd(19932, 468)$$

$$F(5, 3) = 12$$

Problem D. XOR Permutations

Balloon Color: Fuchsia

A bitwise XOR takes two bit patterns of equal length and performs the logical exclusive OR operation on each pair of corresponding bits. The result in each position is 1 if both bits are different (only one of them is 1 and the other is 0), but will be 0 if both bits are the same (both are 0 or both are 1). For example, the bitwise XOR of the patterns 0101 and 1100 is 1001.

In this problem, you are given 3 binary strings of lengths 10 digits such that all digits are either 0 or 1. You can swap any two digits in the same string infinitely, but you cannot swap two digits from two different strings.

Your task is to rearrange digits in the given strings in a way such that the bitwise XOR of the strings after rearranging their digits is as largest as possible. Can you?

Input

The first line contains an integer T ($1 \leq T \leq 250$) specifying the number of test cases.

Each test case consists of 3 lines each of which contains a binary string of length 10 digits such that all digits are either 0 or 1.

Output

For each test, print a single line containing a binary string of length 10 representing the largest value of bitwise XOR that can be obtained by rearranging digits in each string.

A binary string x is larger than a binary string y if after converting both strings to the decimal representation, the decimal value of string x is larger than the decimal value of string y . For example, string "1100" is larger than string "0101" because its decimal value 12, while the decimal value of string "0101" is 5.

Example

Sample Input	Sample Output
2 0000101011 0001010101 0010010000 1000000010 0001000100 1001000000	1111111111 1111110000

Note

In the first test case, you can rearrange the given strings as follow:

- "0000101011" → "0000111100"
- "0001010101" → "1111000000"
- "0010010000" → "0000000011"

Problem E. Building Strings

Balloon Color: Dark Orange

You are given a string s of length n consisting of lowercase English letters. This string can be used to build other strings. The cost of each letter in s is given by another string c of length n consisting of digits, such that the cost of using the letter s_i is c_i coins.

Also, you are given another string p of length m consisting of **unique** lowercase English letters. Your task is to find the minimum cost to build string p by using the letters of s . Can you?

Input

The first line contains an integer T ($1 \leq T \leq 500$) specifying the number of test cases.

The first line of each test case contains two integers n and m ($1 \leq n \leq 10^3$, $1 \leq m \leq 26$), in which n is the length of strings s and c , and m is the length of string p .

Then 3 lines follow, each line contains a string, giving the string s , c , and p , respectively. Both strings s and p contains only lowercase English letters, while string c contains only digits. Also, string p is consisting of **unique** letters.

Output

For each test case, print a single line containing the minimum cost of building the string p by using the letters of string s . If string p cannot be built using string s , print -1 .

Example

Sample Input	Sample Output
3	4
4 2	-1
abcd	8
1234	
ac	
4 4	
abcd	
1234	
abec	
5 3	
abcba	
24513	
acb	

Note

In the first test case, you have to use the 1st and 3rd letters of string s to build string p . So, the total cost is $1 + 3 = 4$.

In the second test case, you cannot build string p using s because the letter 'e' from p does not exist in s . So, the answer is -1 .

In the third test case, the optimal way is to use the 1st, 3rd, and 4th letters of string s to build p . So, the total cost is $2 + 5 + 1 = 8$.

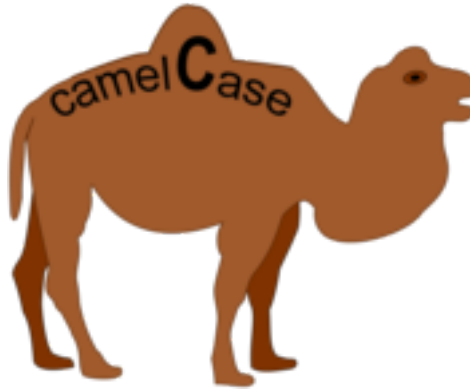
Problem F. camelCase;

Balloon Color: Pink

Camel case is a naming convention practice for multi-word identifiers in programming languages. In this practice, each word except for the first word will begin with an uppercase letter, with no intervening spaces or punctuation.

For example, “problem”, “contestName”, and “contestStartingTime” are all following the camel case practice, while “Server”, “ProblemName”, and “first-Place” are not.

in this problem, you are given a variable name that is following the camel case naming practice, and your task is to determine if the variable name is accepted or not. A variable name is accepted if it is consisting of no more than 7 words.



Input

The first line contains an integer T ($1 \leq T \leq 100$) specifying the number of test cases.

Each test case consists of a single line containing a string s of length no more than 100, giving a variable name. It is guaranteed that the variable name is following the camel case practice as described in the statement, and it contains only lowercase and uppercase English letters.

Output

For each test case, print a single line containing “YES” (without quotes) if the given variable name is accepted. Otherwise, print “NO” (without quotes).

Example

Sample Input	Sample Output
2	YES
weFoundYou	NO
isTheCamelCaseTheBestWayToNameAVariable	

Note

In the first test case, the variable name consists of 3 words, which are: “we”, “Found”, and “You”. So, this variable name is accepted.

In the second test case, the variable name consists of 11 words, which are: “is”, “The”, “Camel”, “Case”, “The”, “Best”, “Way”, “To”, “Name”, “A”, “Variable”. So, this variable name is not accepted.

Problem G. The Special King

Balloon Color: Red

Assem bought a new chess board that has a special king that move in a different way than other regular kings.

In one move, the special king can move from its position in one of the following directions: up, down, left, or right. Formally, if the special king is standing on position (x, y) , in one move it can go to one of the following positions: $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$, or $(x, y + 1)$.

Initially, the special king is standing on position (x_1, y_1) and Assem wants to place it on position (x_2, y_2) . Can you help Assem by calculating the minimum number of required moves he needs to accomplish his goal?

Input

The first line contains an integer T ($1 \leq T \leq 4096$) specifying the number of test cases,

Each test consists of a single line containing four integers x_1, y_1, x_2 , and y_2 ($1 \leq x_1, y_1, x_2, y_2 \leq 8$), in which x_1 and y_1 are representing the starting position of the special king, and x_2 and y_2 are representing the the ending position.

Output

For each test case, print a single line containing the minimum number of required moves to move the special king from the starting position to the ending position.

Example

Sample Input	Sample Output
3	4
1 3 4 2	8
5 7 3 1	0
3 2 3 2	

Note

In the first test case, the special king needs to be moved from position $(1, 3)$ to $(4, 2)$. One possible solution is to make 3 moves down to position $(4, 3)$, then make 1 move to the left to position $(4, 2)$. So, the total number of moves is 4.

Problem H. The Universal String

Balloon Color: Turquoise

Have you ever heard about the universal string? This is the largest string in the world, and it is the mother of all strings that will ever exist in any programming language!

The universal string is an infinite string built by repeating the string “abcdefghijklmnopqrstuvwxyz” infinitely. So, the universal string will look like this:

“...nopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz...”

You are given a string p and your task is to determine if p is a substring of the universal string. Can you?

A **substring** of s is a non-empty string $x = s[l \dots r] = s_l s_{l+1} \dots s_r$ ($1 \leq l \leq r \leq |s|$). For example, “code” and “force” are substring of “codeforces”, while “coders” is not.

Input

The first line contains an integer T ($1 \leq T \leq 10^3$) specifying the number of test cases.

Each test consists of a single line containing a non-empty string p of length no more 10^3 and consisting only of lowercase English letters.

Output

For each test case, print “YES” (without quotes) if the given string is a substring of the universal string. Otherwise, print “NO” (without quotes).

Example

Sample Input	Sample Output
3	YES
abcde	NO
abd	YES
wxyzabc	

Problem I. Array Negations

Balloon Color: Yellow

You are given an array a of n integers, and an integer k . You have to make k negation operations such that at each operation you need to choose an element a_i from the array and replace it with $-a_i$.

Your task is to find the optimal way to make the k negation operations such that at the end the sum of the array a is as maximal as possible. Can you?

Input

The first line contains an integer T ($1 \leq T \leq 100$) specifying the number of test cases.

The first line of each test case contains two integers n and k ($1 \leq n \leq 10^4$, $0 \leq k \leq 10^4$), in which n is the size of the array, and k is the number of negation operations to be made.

Then a line follows contains n integers $a_1, \dots, a_n$ ($-100 \leq a_i \leq 100$), giving the array a .

Output

For each test case, print a single line containing the maximum sum of array a after making the required number of negation operations.

Example

Sample Input	Sample Output
3	8
3 1	4
4 6 2	17
4 2	
-1 0 2 1	
5 2	
1 7 -4 2 -3	

Note

In the first test case, the optimal way is to make the negation operation on a_3 . After this, the array will be $= [4, 6, -2]$, and its sum is 8.

Problem J. Grid Beauty

Balloon Color: Dark Green

You are given a grid g consisting of n rows each of which is divided into m columns.

You can swap any two integers in the same row infinitely, but you cannot swap two integers from two different rows.

Your task is to maximize the beauty of the grid by rearranging integers in each row. The beauty of the grid is the number of pairs (i, j) ($1 < i \leq n$, $1 \leq j \leq m$) such that $g_{i,j}$ is equal to $g_{i-1,j}$ (i.e. $g_{i,j} \equiv g_{i-1,j}$).

Input

The first line contains an integer T ($1 \leq T \leq 5$) specifying the number of test cases.

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 10^3$), giving the number of rows and columns in the grid, respectively.

Then n lines follow, each line contains m integers, giving the grid. All values in the grid are between 1 and 10^8 (inclusive).

Output

For each test case, print a single line containing the beauty of the grid.

Example

Sample Input	Sample Output
2 2 3 1 2 3 4 1 2 3 3 5 7 9 3 2 9 5 3 2	2 3

Note

As input/output can reach huge size it is recommended to use fast input/output methods: for example, prefer to use `scanf/printf` instead of `cin/cout` in C++, prefer to use `BufferedReader/PrintWriter` instead of `Scanner/System.out` in Java.

Problem K. Subarrays OR

Balloon Color: Silver

A bitwise OR takes two equal-length binary representations and performs the logical OR operation on each pair of the corresponding bits. The result in each position is 0 if both bits are 0, while otherwise, the result is 1. The bitwise OR operation is represented in this problem, and also in programming languages such as C++ and Java, using the operator “|”.

To get the value of $x | y$, consider both numbers in the binary representation (padded with zeros to make their lengths equal), then apply the OR operation on the corresponding bits, and return the result into decimal form. For example, the result of $10 | 17 = 01010 | 10001 = 11011 = 27$.

You are given an array a of n integers. A subarray of the array a is a sequence $a_l, a_{l+1}, \dots, a_r$ for some integers (l, r) such that $1 \leq l \leq r \leq n$.

The subarray OR is defined as the bitwise OR of all elements inside that subarray. Formally, the subarray OR of the subarray (l, r) is $a_l | a_{l+1} | \dots | a_r$.

Your task is to find the subarrays OR values of all subarrays in the given array and return the number of **unique** values among them. Can you?

Input

The first line contains an integer T ($1 \leq T \leq 5$) specifying the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 10^5$) giving the size of an array a . Then a line follows contains n integers $a_1, \dots, a_n$ ($0 \leq a_i \leq 10^9$), giving the array a .

Output

For each test case, print a single line containing the number of **unique** subarrays OR values among the subarray OR values of all subarrays in the given array.

Example

Sample Input	Sample Output
2	3
3	6
1 2 3	
3	
2 4 8	

Note

In the first test case, there are 6 subarrays in the given array a . The subarray OR of these subarrays is calculated as follows:

1. $(1, 1) \rightarrow 1$.
2. $(1, 2) \rightarrow 1 | 2 = 3$.
3. $(1, 3) \rightarrow 1 | 2 | 3 = 3$.
4. $(2, 2) \rightarrow 2$.
5. $(2, 3) \rightarrow 2 | 3 = 3$.
6. $(3, 3) \rightarrow 3$.

So, the number of **unique** values among all subarrays is 3.

Problem L. Median

Balloon Color: Light Purple

You are given a grid g consisting of n rows each of which is divided into m columns. The grid contains unique values between 1 and $n \times m$ (inclusive).

Also, you are given two values h and w , and your task is to find the smallest median value that can be found in a rectangle of size $h \times w$.

A rectangle of size $h \times w$ is a rectangle consisting of h rows and w columns such that its top left corner is at a cell (a, b) and its right bottom corner is at a cell (c, d) , and the following conditions are satisfied: $(c - a + 1 \equiv h)$ and $(d - b + 1 \equiv w)$.

The median of a set of numbers is the middle element in the sorted list of the given set. For example, the median of $\{2, 1, 3\}$ is 2 and the median of $\{4, 2, 3, 1, 5\}$ is 3.

Input

The first line contains four integers n , m , h , and w ($1 \leq n, m \leq 10^3$, $1 \leq h, w \leq n$), in which n and m are the number of rows and columns in the grid, respectively, and h and w are representing the size of the required rectangle. Both h and w are odd integers.

Then n lines follow, each line contains m integers, giving the grid. It is guaranteed that the grid contains unique values between 1 and $n \times m$ (inclusive).

Output

Print a single line containing the smallest median value that can be found in a rectangle of size $h \times w$.

Example

Sample Input	Sample Output
4 4 3 3 13 16 15 4 5 2 8 9 14 11 12 10 1 6 3 7	6

Note

In the first test case, there are 4 possible rectangles of size 3×3 , which are:

1. Starts at $(1, 1)$. $\rightarrow$ Contains elements: $\{13, 16, 15, 5, 2, 8, 14, 11, 12\}$. $\rightarrow$ Median value = 12.
2. Starts at $(1, 2)$. $\rightarrow$ Contains elements: $\{16, 15, 4, 2, 8, 9, 11, 12, 10\}$. $\rightarrow$ Median value = 10.
3. Starts at $(2, 1)$. $\rightarrow$ Contains elements: $\{5, 2, 8, 14, 11, 12, 1, 6, 3\}$. $\rightarrow$ Median value = 6.
4. Starts at $(2, 2)$. $\rightarrow$ Contains elements: $\{2, 8, 9, 11, 12, 10, 6, 3, 7\}$. $\rightarrow$ Median value = 8.

So, the smallest median value that can be found in a rectangle of size 3×3 is 6.