

Problema A

Alocação de Prêmios

Uma competição de programação se dará na Nlogônia, para determinar quem é o melhor programador Nlogoniano de todos os tempos.

A competição terá N participantes e não há empates, ou seja, todo participante terminará colocado em uma posição de 1 a N , e todas as posições são distintas. Menores posições correspondem a melhores resultados.

Os organizadores do evento decidiram que cada participante receberá um prêmio de no máximo R pontos e, para ser justo com os participantes com melhor desempenho, um participante nunca receberá menos pontos que um outro com pior colocação.

Alguns participantes, entretanto, são gananciosos e querem receber mais pontos para serem felizes. Um competidor na colocação i precisa receber um prêmio de pelo menos p_i pontos para ser feliz.

Ina, uma organizadora muito curiosa, está se perguntando de quantas maneiras é possível distribuir prêmios aos participantes de maneira a satisfazer as condições da organização, e fazer todos os participantes felizes. Como esse número é muito grande, você deve calculá-lo módulo $10^9 + 7$.

Duas maneiras são diferentes se pelo menos um competidor recebe um prêmio diferente.

Entrada

A primeira linha contém dois inteiros N e R ($1 \leq N \leq 5000$, $1 \leq R \leq 10^9$), representando o número de participantes e o prêmio máximo que cada competidor pode receber, respectivamente.

A segunda linha contém N inteiros, p_i ($1 \leq p_i \leq 10^9$), representando a menor quantidade de pontos que o participante na colocação i precisa receber para ficar feliz.

Saída

Imprima uma linha contendo o número de diferentes maneiras de se distribuir o prêmio, módulo $10^9 + 7$.

Exemplo de entrada 1 2 5 4 1	Exemplo de saída 1 9
Exemplo de entrada 2 3 10 7 1 10	Exemplo de saída 2 1

Problem A

Assigning Prizes

A programming competition will be held in Nlogonia to determine who is the best Nlogonian programmer of all time.

The competition will have N contestants and there are no ties, in other words, every contestant will be ranked in a position from 1 to N and all ranks are distinct. Lower ranks represent better results.

The event organizers decided that each contestant will receive a prize of at most R rating points, and, to be fair to the competitors that performed better, a contestant will never receive fewer rating points than any other contestant with a worse ranking.

Some competitors, though, are more greedy and want to receive more rating points to be happy. A contestant with rank i needs to receive a prize of at least p_i rating points to be happy.

Ina, a very curious organizer, is wondering in how many ways it is possible to distribute the prizes to the contestants in order to satisfy the organization's conditions and make all contestants happy. Also, since this number can be very large, you should calculate it modulo $10^9 + 7$.

Two ways are different if at least one contestant receives a different prize amount.

Input

The first line contains two integers N and R ($1 \leq N \leq 5000$, $1 \leq R \leq 10^9$), representing the number of contestants and the maximum amount of rating points that each contestant can receive as a prize, respectively.

The second line contains N integers, p_i ($1 \leq p_i \leq 10^9$), representing the minimum rating points the contestant ranked i needs to receive as a prize to be happy.

Output

Print the number of different ways to distribute the prizes modulo $10^9 + 7$.

Input example 1 2 5 4 1	Output example 1 9
Input example 2 3 10 7 1 10	Output example 2 1

Problema A

Asignación de premios

Pronto se realizará una competencia en Nlogonia para determinar quién es el mejor programador Nlogonio de todos los tiempos.

Participarán N competidores y no habrá empates. Es decir, cada competidor terminará ubicado en una posición diferente desde 1 hasta N inclusive. Una menor posición indica un resultado mejor.

Los organizadores del evento han decidido otorgar a cada competidor un premio, en forma de puntos de rating. Como máximo, cada competidor puede recibir R puntos de rating. Para ser justos con los competidores que tuvieron un mejor desempeño, un competidor nunca recibirá menos puntos de rating que otro competidor con una posición peor.

Algunos competidores sin embargo son más codiciosos, y quieren recibir más puntos de rating para estar contentos. El competidor ubicado en la posición i necesita recibir un premio de al menos p_i puntos de rating para estar contento.

Ina es una organizadora muy curiosa, y se pregunta de cuántas maneras diferentes es posible asignar los premios a los competidores, de modo tal que se satisfagan las condiciones impuestas por la organización y que todos los competidores estén contentos. Además, como este número puede ser muy grande, debes calcularlo módulo $10^9 + 7$.

Dos maneras de asignar premios son diferentes, si al menos un competidor recibe una cantidad diferente de puntos de rating.

Entrada

La primera línea contiene dos enteros N y R ($1 \leq N \leq 5000$, $1 \leq R \leq 10^9$), que indican la cantidad de competidores y la cantidad máxima de puntos de rating que puede recibir cada competidor como premio, respectivamente.

La segunda línea contiene N enteros, p_i ($1 \leq p_i \leq 10^9$), que indican la mínima cantidad de puntos de rating que cada competidor necesita recibir como premio para estar contento.

Salida

Imprima la cantidad de formas diferentes en que se pueden asignar los puntos de rating de premio, módulo $10^9 + 7$.

Ejemplo de entrada 1 2 5 4 1	Ejemplo de salida 1 9
Ejemplo de entrada 2 3 10 7 1 10	Ejemplo de salida 2 1

Problema B

Belas Palavras

São dados uma string A de comprimento N e um conjunto S contendo M strings.

Uma permutação cíclica B_i de A , onde i é um número entre 1 e N , é a string

$$B_i = A_i A_{i+1} \cdots A_{N-1} A_N A_1 A_2 \cdots A_{i-2} A_{i-1}$$

e a sua pontuação é definida como o maior comprimento de uma substring de B_i que é também uma substring de uma string em S .

Uma substring é definida como uma sequência contígua de letras. Por exemplo, **ab** e **dc** são substrings de **abfdc**, mas **ad** e **fc** não são substrings de **abfdc**.

Sua tarefa é calcular a menor pontuação dentre todas as permutações cíclicas da string A .

Entrada

A primeira linha contém dois inteiros positivos N e M , ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^4$), representando o comprimento da string A e o tamanho do conjunto S , respectivamente.

A segunda linha contém a string A .

Cada uma das M linhas seguintes contém uma string s_i , representando a i -ésima string em S .

Todas as strings contêm apenas letras minúsculas do alfabeto, e é garantido que a soma dos tamanhos de todas as strings em S nunca ultrapassa 10^5 caracteres.

Saída

Imprima uma linha contendo um inteiro representando a menor pontuação dentre todas as permutações cíclicas da string A .

Exemplo de entrada 1 7 3 acmicpc acm icpc maratona	Exemplo de saída 1 3
Exemplo de entrada 2 11 4 competition oncom petition ztxvu fmwper	Exemplo de saída 2 5
Exemplo de entrada 3 12 4 latinamerica zyvu okp wsgh kqpdb	Exemplo de saída 3 0

Problem B

Beautiful Words

You are given a string A of length N and a set S , containing M strings.

A cyclic permutation B_i of A , in which i is between 1 and N , is the string

$$B_i = A_i A_{i+1} \cdots A_{N-1} A_N A_1 A_2 \cdots A_{i-2} A_{i-1}$$

and its score is defined as the maximum length of a substring of B_i that is also a substring of some string in S .

A substring is defined as a contiguous sequence of letters. For example, **ab** and **dc** are substrings of **abfdc**, but **ad** and **fc** aren't substrings of **abfdc**.

Your task is to calculate the minimum score over all cyclic permutations of string A .

Input

The first line contains two positive integers N and M , ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^4$), representing the length of the string A and the size of the set S , respectively.

The second line contains the string A .

Each of the next M lines contains one string s_i , representing the i -th string in S .

All strings contain only lowercase English letters and it's guaranteed that the sum of lengths of all strings in S never exceeds 10^5 characters.

Output

Output an integer representing the minimum score over all cyclic permutations of string A .

Input example 1 7 3 acmicpc acm icpc maratona	Output example 1 3
Input example 2 11 4 competition oncom petition ztxvu fmwper	Output example 2 5
Input example 3 12 4 latinamerica zyvu okp wsgh kqpdb	Output example 3 0

Problema B

Bellas palabras

Dada una cadena A de longitud N y un conjunto S de M cadenas.

Una permutación cíclica B_i de A , en la que i se encuentra entre 1 y N , es la cadena:

$$B_i = A_i A_{i+1} \cdots A_{N-1} A_N A_1 A_2 \cdots A_{i-2} A_{i-1}$$

Y su puntuación se define como la longitud máxima de una subcadena de B_i que también es subcadena de alguna cadena en S .

Una subcadena está definida como una secuencia contigua de letras. Por ejemplo, **ab** y **dc** son subcadenas de **abfdc**, pero, **ad** y **fc** no son subcadenas de **abfdc**.

Su tarea es calcular el puntaje mínimo sobre todas las permutaciones cíclicas de la cadena A .

Entrada

La primera línea contiene dos enteros positivos N y M , ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^4$), indicando, respectivamente, la longitud de la cadena A y el tamaño del conjunto S .

La segunda línea contiene la cadena A .

Cada una de las siguientes M líneas contiene una cadena s_i , representando la i -ésima cadena del conjunto S .

Todas las cadenas contienen solo letras minúsculas del alfabeto inglés. Se garantiza que la suma de las longitudes de las cadenas en S nunca será mayor a 10^5 caracteres.

Salida

Imprima una línea con un número entero, indicando la mínima puntuación sobre todas las permutaciones cíclicas de A .

Ejemplo de entrada 1 7 3 acmicpc acm icpc maratona	Ejemplo de salida 1 3
Ejemplo de entrada 2 11 4 competition oncom petition ztxvu fmwper	Ejemplo de salida 2 5
Ejemplo de entrada 3 12 4 latinamerica zyvu okp wsgh kqpdb	Ejemplo de salida 3 0

Problema C

Criando Múltiplos

Malba é um garoto muito inteligente que gosta de calcular. Já ganhou muitas competições, inclusive a prestigiosa competição Tahan, em que conseguiu o primeiro lugar, representando o seu país, a Logônia.

Ele inventou um problema, no qual ele considera um número N , escrito numa certa base B , e representado por L algarismos. O objetivo do jogo é reduzir não mais do que um dos algarismos de forma que o novo número, M , seja um múltiplo do número $B + 1$. Mas há um detalhe: dentre as alterações possíveis, deve-se escolher uma que minimize M .

Por exemplo, suponha que $B = 10$ e $N = 23456$. Há duas maneiras de obter M : ou reduzimos o algarismo 4 para 0 ou reduzimos o algarismo 6 para 2. Então, o 4 deve ser reduzido para 0, portanto $M = 23056$. Em alguns casos não há solução, como no caso em que $B = 10$ e $N = 102$. Nesse caso, se trocarmos o algarismo 1 por 9 obteremos um múltiplo de 11, mas não podemos aumentar o valor de um algarismo!

Observe que pode ser necessário reduzir o primeiro algarismo para o valor 0. Por exemplo, isto acontece se $B = 10$ e $N = 322$.

Você consegue dizer qual dígito deve ser reduzido e qual seu novo valor?

Entrada

A primeira linha contém dois inteiros B e L ($2 \leq B \leq 10^4$, $1 \leq L \leq 2 \times 10^5$), representando a base e o número de algarismos do número N , respectivamente.

A segunda linha contém L inteiros D_1, D_2, \dots, D_L ($0 \leq D_i < B$ para $i = 1, 2, \dots, L$), representando os algarismos do número N . O primeiro algarismo, D_1 , é o mais significativo e o último algarismo, D_L , é o menos significativo.

Saída

Imprima uma linha contendo dois inteiros, separados por um espaço. O primeiro inteiro é o índice do algarismo a ser alterado (lembre que o índice do primeiro algarismo, D_1 , é 1 e o índice do último algarismo, D_L , é L). O segundo inteiro é o novo valor do algarismo. Se não houver solução para o problema, imprima -1 -1. Se N já for um múltiplo de $B + 1$, imprima 0 0.

Exemplo de entrada 1 10 5 2 3 4 5 6	Exemplo de saída 1 3 0
Exemplo de entrada 2 10 3 1 0 2	Exemplo de saída 2 -1 -1
Exemplo de entrada 3 2 5 1 0 1 1 1	Exemplo de saída 3 4 0
Exemplo de entrada 4 17 5 3 0 0 0 0	Exemplo de saída 4 1 0

Exemplo de entrada 5 16 4 15 0 13 10	Exemplo de saída 5 1 14
Exemplo de entrada 6 16 5 1 15 0 13 10	Exemplo de saída 6 0 0

Problem C

Creating Multiples

Malba is a very smart kid who loves to perform calculations. He has won several competitions, including the prestigious Tahan competition, in which he got the first prize, representing his country, Logonia.

He created a puzzle, in which he considers a number N , written in a certain base B , and represented by L digits. The objective of the game is to reduce at most one of the digits so that the new number, M , be a multiple of the number $B + 1$. But there is a catch: among the possible solutions you must choose one that renders M the smallest possible value.

For example, suppose that $B = 10$ and $N = 23456$. There are two ways in which M may be obtained: either we reduce the digit 4 to 0 or we reduce the digit 6 to 2. Thus, 4 must be changed to 0, hence $M = 23056$. Sometimes there is no solution, as is the case if $B = 10$ and $N = 102$. In this case, if we change the digit 1 to 9 we get a multiple of 11, but we are not allowed to increase the value of a digit!

Observe that it may be necessary to reduce the first digit to 0. For example, this is the case if $B = 10$ and $N = 322$.

Can you tell which digit should be reduced and what is its new value?

Input

The first line contains two integers B and L ($2 \leq B \leq 10^4$, $1 \leq L \leq 2 \times 10^5$), representing the base and the number of digits of the number N , respectively.

The second line contains L integers D_1, D_2, \dots, D_L ($0 \leq D_i < B$ for $i = 1, 2, \dots, L$), representing the digits of the number N . The first digit, D_1 , is the most significant and the last, D_L , is the least significant.

Output

Output a line containing two integers, separated by one space. The first integer is the index of the digit to be changed (recall that the index of the first digit, D_1 , is 1 and the index of the last digit, D_L , is L). The second integer is the new value of the digit. If there is no solution to the problem, output -1 -1. If N is already a multiple of $B + 1$ then output 0 0.

Input example 1 10 5 2 3 4 5 6	Output example 1 3 0
Input example 2 10 3 1 0 2	Output example 2 -1 -1
Input example 3 2 5 1 0 1 1 1	Output example 3 4 0
Input example 4 17 5 3 0 0 0 0	Output example 4 1 0

Input example 5 16 4 15 0 13 10	Output example 5 1 14
Input example 6 16 5 1 15 0 13 10	Output example 6 0 0

Problema C

Creando múltiplos

Malba es un niño muy inteligente al que le gusta realizar cálculos. Ha ganado varias competencias, incluida la prestigiosa competencia Tahan, donde obtuvo el primer lugar representando a su país, Logonia.

Ha creado un acertijo, en el cual considera un número N que al escribirse en una cierta base B , está formado por L dígitos. En el acertijo se debe reducir a lo sumo uno de los dígitos, de tal modo que se forme un nuevo número M que sea múltiplo de $B + 1$. Pero, hay una trampa: si hay varias opciones, deberá ser alterado el dígito que haga el valor de M lo más bajo posible.

Por ejemplo, supongamos que $B = 10$ y $N = 23456$. Hay dos maneras en las que se puede obtener M : ya sea que se reduzca el dígito 4 a 0, o que se reduzca el dígito 6 a 2. Entonces, 4 debe ser cambiado a 0, y entonces $M = 23056$. A veces no hay solución alguna, como es el caso si $B = 10$ y $N = 102$. En este caso, si se cambiara el dígito 1 por el 9 obtendríamos un múltiplo de 11, ¡pero no tenemos permitido incrementar el valor de un dígito!

Obsérvese que puede ser necesario reducir el primer dígito a 0. Por ejemplo cuando $B = 10$ y $N = 322$.

¿Puedes decir cuál es el dígito que se debe reducir, y cuál es su nuevo valor?

Entrada

La primera línea contiene dos enteros B y L ($2 \leq B \leq 10^4$, $1 \leq L \leq 2 \times 10^5$), que representan respectivamente, la base y la cantidad de dígitos del número N .

La segunda línea contiene L enteros D_1, D_2, \dots, D_L ($0 \leq D_i < B$ para $i = 1, 2, \dots, L$), que representan los dígitos del número N . El primer dígito, D_1 , es el más significativo y el último, D_L , es el menos significativo.

Salida

Debes imprimir una línea que contenga dos enteros separados por un espacio. El primer entero es el índice del dígito que debe cambiarse (recuerda que el índice del primer dígito: D_1 , es 1; y el índice del último dígito: D_L es L). El segundo entero es el nuevo valor del dígito. Si no hay una solución al problema, imprime -1, -1. Si N ya es un múltiplo de $B + 1$, imprime 0, 0.

Ejemplo de entrada 1 10 5 2 3 4 5 6	Ejemplo de salida 1 3 0
Ejemplo de entrada 2 10 3 1 0 2	Ejemplo de salida 2 -1 -1
Ejemplo de entrada 3 2 5 1 0 1 1 1	Ejemplo de salida 3 4 0
Ejemplo de entrada 4 17 5 3 0 0 0 0	Ejemplo de salida 4 1 0

Ejemplo de entrada 5 16 4 15 0 13 10	Ejemplo de salida 5 1 14
Ejemplo de entrada 6 16 5 1 15 0 13 10	Ejemplo de salida 6 0 0

Problema D

Dividindo o Reino

O reino da Nlogônia historicamente tem sido um lugar rico e tranquilo. Entretanto, as atuais circunstâncias podem dar fim a esta era de paz e prosperidade: o rei é pai de gêmeos, então ambos são herdeiros do trono.

Os gêmeos não se dão muito bem e são ciumentos e competitivos um com o outro. Devido a este fato, tê-los governando o reino cooperativamente não é realmente uma opção. O reino terá de ser dividido em dois principados independentes, de forma que cada um seja dado a um príncipe. Além disso, a divisão precisa ser totalmente justa, para evitar conflitos entre os irmãos invejosos.

O reino consiste em N cidades e M estradas conectando pares de cidades. Os Nlogonianos são particularmente orgulhosos de suas estradas. Cada estrada tem um valor positivo associado a ela, representando sua beleza.

O reino será dividido da seguinte maneira: primeiro, as cidades serão particionadas em dois conjuntos de forma que cada cidade está em exatamente um conjunto. Então, cada principado será composto pelas cidades em um dos conjuntos e pelas estradas conectando as cidades naquele conjunto. Estradas que conectam cidades de diferentes principados serão destruídas, uma vez que os príncipes não estão interessados em fazer negócios ou colaborações entre os principados, e manter estradas apenas aumentaria a chance de guerras.

A beleza de um principado é definida como a maior beleza dentre todas as estradas daquele principado, ou 0 (zero) se o principado não possui estradas. Por motivos óbvios, o rei gostaria que a beleza de ambos os principados fosse igual.

Ajude o rei a determinar todos os possíveis valores de beleza dos possíveis principados resultantes, dado que a divisão é feita de maneira que os principados sejam igualmente belos.

Entrada

A primeira linha contém dois inteiros N, M ($1 \leq N, M \leq 5 \times 10^5$), representando o número de cidades e o número de estradas respectivamente.

Cada uma das M linhas seguintes contém três inteiros x_i, y_i, b_i ($1 \leq x_i < y_i \leq N, 1 \leq b_i \leq 10^9$), representando que existe uma estrada que conecta as cidades x_i e y_i e possui beleza b_i . Não existem estradas entre um mesmo par de cidades.

Saída

Se não for possível dividir o reino de maneira que ambos os principados tenham a mesma beleza, imprima uma linha com a string “IMPOSSIBLE”. Caso contrário, imprima todos os possíveis valores resultantes da divisão do reino em principados de mesma beleza. Os valores devem ser impressos em ordem crescente, um por linha.

Exemplo de entrada 1	Exemplo de saída 1
9 7	2
1 2 3	3
2 3 3	
3 4 3	
1 3 2	
2 4 2	
6 7 1	
8 9 1	

Exemplo de entrada 2 4 4 1 2 5 2 3 6 1 3 7 3 4 7	Exemplo de saída 2 IMPOSSIBLE
Exemplo de entrada 3 2 1 1 2 10	Exemplo de saída 3 0

Problem D

Dividing the Kingdom

The kingdom of Nlogonia has historically been a very wealthy and quiet place. However, the current circumstances could bring this era of peace and prosperity to an end: The king is a father of two twins, so both of them are heirs to the throne.

The twins don't get along well and are jealous and overly competitive towards each other. Due to this, having both of them rule over the kingdom cooperatively is not a viable option. The kingdom will have to be divided into two independent principalities so that each of them can be given to each prince. Also, the division needs to be totally fair to avoid conflict between the envious brothers.

The kingdom consists of N cities and M roads connecting pairs of cities. The Nlogonians are peculiarly proud of their roads. Each road has an associated positive value which represents its beauty.

The kingdom will be divided in this manner: First, the cities will be partitioned in two sets such that every city is in one and only one set. Then, each principality will consist of the cities in one set and the roads connecting cities of this same set. Roads that connect cities of different principalities will be destroyed, as the princes are not interested in trading with each other, and keeping the roads would only make war more likely.

The beauty of a principality is defined as the maximum beauty of the roads within the principality, or 0 (zero) if the principality has no roads at all. For obvious reasons, the king would like the beauty of both principalities to be the same.

Help the king determine all the possible values of the beauty of the resulting principalities, given that the division is made in such a way that the principalities are equally beautiful.

Input

The first line contains two integers N, M ($1 \leq N, M \leq 5 \times 10^5$), representing the number of cities and the number of roads respectively.

Each of the next M lines contains three integers x_i, y_i, b_i ($1 \leq x_i < y_i \leq N, 1 \leq b_i \leq 10^9$), representing that there's a road which connects cities x_i and y_i and has beauty b_i . There's no two roads connecting the same pair of cities.

Output

If it's not possible to divide the kingdom so that both principalities have the same beauty, output a line with the string "IMPOSSIBLE". Otherwise, output all the possible values for principality beauty resulting from divisions in principalities with equal beauty. Values should be outputted in ascending order, each in its own line.

Input example 1	Output example 1
9 7	2
1 2 3	3
2 3 3	
3 4 3	
1 3 2	
2 4 2	
6 7 1	
8 9 1	

Input example 2 4 4 1 2 5 2 3 6 1 3 7 3 4 7	Output example 2 IMPOSSIBLE
Input example 3 2 1 1 2 10	Output example 3 0

Problema D

Dividiendo el reino

El reino de Nlogonia ha sido históricamente un lugar muy rico y tranquilo. Sin embargo, las circunstancias actuales podrían llevar esta era de paz y prosperidad a su fin: El rey es padre de dos mellizos, por lo que ambos son herederos al trono.

Los mellizos no se llevan bien y son celosos y excesivamente competitivos uno con otro. Debido a esto, hacer que ambos gobiernen el reino cooperativamente no es una opción viable. El reino tendrá que ser dividido en dos principados independientes, de modo que cada una de ellos sea dado a cada príncipe. Además, la división debe ser totalmente justa para evitar conflicto entre los envidiosos hermanos.

El reino consta de N ciudades y M caminos que conectan pares de ciudades. Los nlogonios son particularmente orgullosos de sus caminos. Cada camino tiene asociado un valor positivo que representa su belleza.

El reino será dividido de este modo: Primero, las ciudades serán particionadas en dos conjuntos tales que cada ciudad está en uno y sólo un conjunto. Entonces, cada principado constará de las ciudades en un conjunto y los caminos conectando ciudades de este mismo conjunto entre sí. Los caminos que conectan ciudades de distintos principados serán destruidos, ya que los príncipes no están interesados en comerciar uno con otro y mantener los caminos sólo incrementaría la probabilidad de guerra.

La belleza de un principado se define como la mayor belleza entre los caminos del principado, o 0 (cero) si el principado no tiene ningún camino. Por razones obvias, el rey quiere que la belleza de ambos principados sea la misma.

Ayuda al rey a determinar todos los posibles valores de la belleza de los principados resultantes, dado que la división es hecha de tal manera que los principados son igual de bellos.

Entrada

La primera línea contiene dos enteros N, M ($1 \leq N, M \leq 5 \times 10^5$), que representan la cantidad de ciudades y la cantidad de caminos, respectivamente.

Cada una de las siguientes M líneas contiene tres enteros x_i, y_i, b_i ($1 \leq x_i < y_i \leq N, 1 \leq b_i \leq 10^9$), que representan que hay un camino que conecta las ciudades x_i e y_i y tiene belleza b_i . No hay dos caminos conectando el mismo par de ciudades.

Salida

Si no es posible dividir el reino de modo que ambos principados tengan la misma belleza, imprime una línea con la cadena “IMPOSSIBLE”. En caso contrario, imprime todos los posibles valores para la belleza de los principados resultantes de divisiones en principados con igual belleza. Los valores deben ser impresos en orden ascendente, cada uno en su propia línea.

Ejemplo de entrada 1	Ejemplo de salida 1
9 7	2
1 2 3	3
2 3 3	
3 4 3	
1 3 2	
2 4 2	
6 7 1	
8 9 1	

Ejemplo de entrada 2 4 4 1 2 5 2 3 6 1 3 7 3 4 7	Ejemplo de salida 2 IMPOSSIBLE
Ejemplo de entrada 3 2 1 1 2 10	Ejemplo de salida 3 0

Problema E

Escada Rolante

Você acaba de inventar um novo tipo de escada rolante: uma escada rolante dupla. Escadas rolantes normais levam as pessoas de uma das pontas para a outra, mas não na direção contrária, enquanto que as escadas rolantes duplas podem levar pessoas de qualquer uma das pontas para a outra.

Leva-se 10 segundos para que a escada rolante dupla leve uma pessoa de uma das pontas até a outra. Isto é, se a pessoa entra na escada rolante dupla em uma das pontas no momento T , então vai sair na outra ponta no momento $T + 10$ – esta pessoa não estará mais na escada rolante dupla no momento $T + 10$.

A todo momento que ninguém esteja usando a escada rolante dupla, ela estará parada. Portanto, inicialmente ela está parada.

Quando a escada rolante dupla está parada e uma pessoa entra por uma das pontas, a escada rolante dupla se ligará automaticamente e se moverá na direção que aquela pessoa quer ir.

Se uma pessoa chegar na escada rolante dupla e esta já estiver movendo-se na direção que a pessoa quer ir, então a pessoa entrará nela imediatamente. Caso contrário, se a escada rolante estiver se movendo na direção oposta, a pessoa terá que esperar até que a escada rolante pare e só então a pessoa poderá entrar nela. A escada rolante dupla é tão larga que ela pode acomodar inúmeras pessoas entrando nela ao mesmo tempo.

A escada rolante dupla tem um efeito bem estranho, provavelmente relacionado a alguma propriedade da física quântica (ou simplesmente ao acaso): nenhuma pessoa vai chegar na escada rolante dupla no momento exato em que ela está prestes a parar.

Agora que você sabe como a escada rolante dupla funciona, você terá a tarefa de simulá-la. Dada a informação de N pessoas, incluindo o momento em que elas chegaram na escada rolante dupla e em qual direção elas querem andar, você tem que descobrir qual o último momento em que a escada para.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 10^4$), representando quantas pessoas usarão a escada rolante.

Em seguida haverão N linhas contendo dois inteiros t_i e d_i cada ($1 \leq t_i \leq 10^5$, $0 \leq d_i \leq 1$), representando o momento em que a i -ésima pessoa chegará na escada rolante dupla e em qual direção ela quer ir. Se d_i é igual a 0, então a pessoa quer ir da ponta esquerda para a ponta direita, e se d_i é igual 1, então a pessoa quer ir da ponta direita para a ponta esquerda. Todos os valores de t_i são distintos e dados em ordem crescente.

Saída

Imprima uma linha contendo o momento no qual a última pessoa saiu da escada rolante.

Exemplo de entrada 1	Exemplo de saída 1
3 5 0 8 0 13 0	23

Exemplo de entrada 2 3 5 0 7 1 9 0	Exemplo de saída 2 29
Exemplo de entrada 3 3 5 0 10 1 16 0	Exemplo de saída 3 35

Problem E

Escalator

You have just invented a new type of escalator: the double escalator. Regular escalators take people from one endpoint to another but not in the other direction, while the double escalator can take people from any one of its endpoints to the other one.

It takes 10 seconds for the double escalator to take a person from any of its endpoints to the other one. That is, if a person enters the double escalator from one of the endpoints at moment T , then they will leave at the other endpoint at moment $T + 10$ – this person won't be using the double escalator anymore at moment $T + 10$.

Any time that no one is using the double escalator, it will stop immediately. Thus, it is initially stopped.

When the double escalator is stopped and a person enters it from one of its endpoints, it will turn on automatically and move in the direction that this person wants to go.

If a person arrives at the double escalator and it is already moving in the direction that they want to go, they will enter it immediately. Otherwise, if it's moving in the opposite direction that they want to go, they will wait until it stops, and only then will they enter it. The escalator is so large that it can accommodate many people entering it at the same time.

The double escalator has a very weird effect, probably related to some quantum physics effect (or just chance): no person will ever arrive on the double escalator at the exact moment the escalator stops.

Now that you know how the double escalator works, you will be given the task of simulating it. Given the information about N people, including their time of arrival at the escalator and which direction they want to go, you have to figure out the last moment that the escalator stops.

Input

The first line contains one integer N ($1 \leq N \leq 10^4$), representing the number of people that will use the double escalator.

Each of the next N lines contains two integers t_i and d_i ($1 \leq t_i \leq 10^5$, $0 \leq d_i \leq 1$), representing the time that the i -th person will arrive at the escalator and which direction they want to go. If d_i is equal to 0, they want to go from the left to the right endpoint, and if d_i is equal to 1, they want to go from the right to the left endpoint. All values of t_i are distinct and will be given in ascending order.

Output

Output one line containing the time that the last person will leave the double escalator.

Input example 1 3 5 0 8 0 13 0	Output example 1 23
Input example 2 3 5 0 7 1 9 0	Output example 2 29

Input example 3	Output example 3
3 5 0 10 1 16 0	35

Problema E

Escalera mecánica

Acabas de inventar un nuevo tipo de escalera mecánica: la escalera mecánica doble. En una escalera mecánica normal, las personas son transportadas desde un extremo hasta el otro, siempre en esa misma dirección. En cambio, la escalera mecánica doble es capaz de transportar personas desde cualquiera de sus extremos hasta el otro.

La escalera mecánica doble se demora 10 segundos en transportar una persona desde cualquiera de los extremos hasta el otro. Es decir, si una persona ingresa a la escalera desde uno de los extremos en el instante T , entonces saldrá por el otro extremo en el instante $T + 10$ – esta persona ya no estará utilizando la escalera en el instante $T + 10$.

En cualquier instante de tiempo en el que nadie esté utilizando la escalera mecánica doble, esta se detiene inmediatamente. La escalera se encuentra inicialmente detenida.

Cuando la escalera se encuentra detenida y una persona ingresa desde uno de los extremos, la escalera se enciende automáticamente y se mueve en la dirección en la que esta persona desea desplazarse.

Si una persona llega a la escalera y esta ya se encuentra en movimiento en la dirección en la que la persona desea desplazarse, la persona ingresa inmediatamente a la escalera. De lo contrario, si la escalera se encuentra moviéndose en la dirección opuesta, la persona esperará hasta que la escalera se detenga, y sólo entonces ingresará. La escalera mecánica es tan grande que tiene lugar para que mucha gente ingrese en ella al mismo tiempo.

La escalera mecánica doble tiene un efecto muy extraño, probablemente relacionado con algún efecto de física cuántica (o quizás sea simplemente por azar): ninguna persona llegará nunca a la escalera en el momento exacto en el que esta se detiene.

Ahora que sabes cómo opera la escalera mecánica doble, tu tarea consiste en simular su funcionamiento. Dada la información acerca de N personas, incluyendo sus tiempos de llegada a la escalera y en qué dirección desean desplazarse, debes calcular en qué momento la escalera se detendrá por última vez.

Entrada

La primera línea contiene un entero N ($1 \leq N \leq 10^4$), que indica la cantidad de personas que utilizarán la escalera mecánica.

Cada una de las siguientes N líneas contiene dos enteros t_i y d_i ($1 \leq t_i \leq 10^5$, $0 \leq d_i \leq 1$), que representan respectivamente el instante de tiempo en el que la i -ésima persona llegará a la escalera, y en qué dirección desea desplazarse. Si d_i es 0, la persona desea desplazarse desde el extremo izquierdo hasta el derecho, y si d_i es 1, la persona desea desplazarse desde el extremo derecho hasta el izquierdo. Todos los valores t_i son distintos, y serán dados en orden ascendente.

Salida

Debes escribir una única línea, que contenga el instante de tiempo en el que la escalera se detendrá por última vez.

Ejemplo de entrada 1	Ejemplo de salida 1
3 5 0 8 0 13 0	23

Ejemplo de entrada 2 3 5 0 7 1 9 0	Ejemplo de salida 2 29
Ejemplo de entrada 3 3 5 0 10 1 16 0	Ejemplo de salida 3 35

Problema F

Fuga da Prisão

Michael e seu irmão Lincoln foram presos injustamente, na mesma prisão, mas Michael tem um plano para resgatar seu irmão. Pode-se considerar que a prisão é um conjunto de polígonos convexos no plano, cujas arestas são muros. Os muros de polígonos distintos não se interceptam, mas os polígonos podem ser encaixados, isto é, um polígono pode estar dentro de outro polígono. Pode-se considerar que Michael e Lincoln são dois pontos no plano. O caminho para o resgate consiste em primeiro Michael chegar até seu irmão e então ambos precisam escapar da prisão.

Eles não têm problema para andar, mas escalar muros é perigoso e difícil, assim Michael tentará minimizar o número total de muros que ele deverá escalar. Portanto, Michael primeiro precisa escalar alguns muros para chegar ao seu irmão, caso não se encontrem na mesma área. Em seguida deve escalar mais alguns muros para deixar a prisão. Deixar a prisão significa não estar dentro de quaisquer muros, que pode-se considerar equivalente a atingir um ponto muito longe, digamos, $(10^{20}, 10^{20})$. Brad está encarregado da alocação dos prisioneiros e está sabendo do plano, de forma que ele colocará os dois prisioneiros em dois pontos diferentes do plano, mas não nos muros, e de tal forma que o número mínimo de muros que precisam ser escalados por Michael seja o maior possível. Qual é o valor do número mínimo de muros que Michael precisará escalar se Brad colocar os dois irmãos de forma ótima?

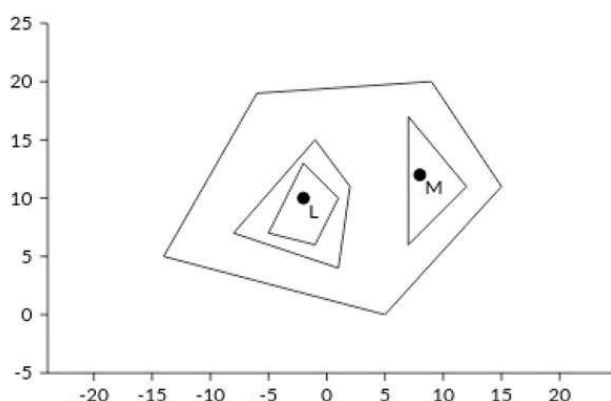


Figura 1: Ilustração de possível solução para exemplo 1. O ponto M representa Michael e o ponto L Lincoln

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 2 \times 10^5$), que é o número de polígonos convexos. Esta linha é seguida pelas descrições de cada polígono. A primeira linha da i -ésima descrição contém um inteiro k_i ($3 \leq k_i \leq 6 \times 10^5$), seguida de k_i linhas, cada uma das quais contém um ponto (x_j, y_j) ($-10^9 \leq x_j, y_j \leq 10^9$).

Os pontos para formar cada polígono convexo são dados na ordem anti horária e não há três pontos consecutivos colineares. As arestas de dois polígonos distintos não se interceptam. O número total de arestas não passa de 6×10^5 , ou seja, $\sum_{i=1}^N k_i \leq 6 \times 10^5$.

Saída

Imprima um inteiro, o número mínimo de muros que precisarão ser escalados por Michael para resgatar seu irmão, supondo que Brad os colocou em lugares que torna esse número de muros o maior possível.

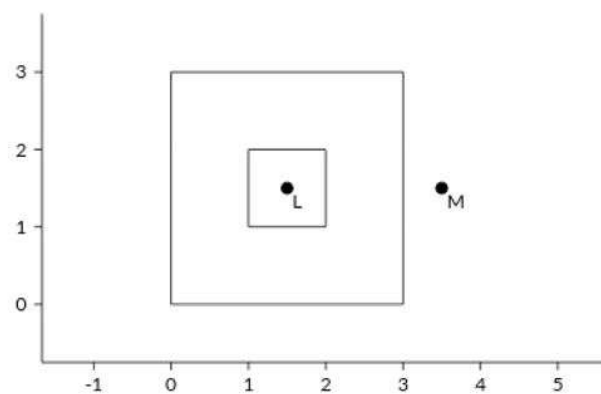


Figura 2: Ilustração de possível solução para exemplo 2

Exemplo de entrada 1	Exemplo de saída 1
4 4 1 10 -2 13 -5 7 -1 6 5 15 11 9 20 -6 19 -14 5 5 0 4 -1 15 -8 7 1 4 2 11 3 7 17 7 6 12 11	6

Exemplo de entrada 2	Exemplo de saída 2
2 4 0 0 3 0 3 3 0 3 4 1 1 2 1 2 2 1 2	4

Problem F

Freedom from Prison

Michael and his brother Lincoln are unfairly imprisoned in the same prison, but Michael has a plan to rescue his brother. The prison can be seen as a set of convex polygons in the plane, in which the edges of the polygons are walls. The walls of distinct polygons do not intersect, but polygons may be nested, that is, inside one another. Michael and Lincoln can be seen as two points in the plane. The rescue path will be for Michael to reach his brother and then both of them need to escape the prison.

Walking for them is no problem, but climbing walls is dangerous and difficult, so Michael will try to minimize the total number of walls climbed by him. So Michael first needs to climb some walls to reach his brother if they are not within the same area and then climb some more walls to leave the prison. Leaving the prison means not being within any walls, which can be seen as reaching a point very far away, let's say $(10^{20}, 10^{20})$. Brad is in charge of the placements of the prisoners and is aware of the plan, so he will place both prisoners in two different points in the plane not contained by any segments and such that the minimum amount of walls that need to be climbed by Michael is maximum. What is the minimum amount of walls to be climbed if Brad places the brothers optimally?

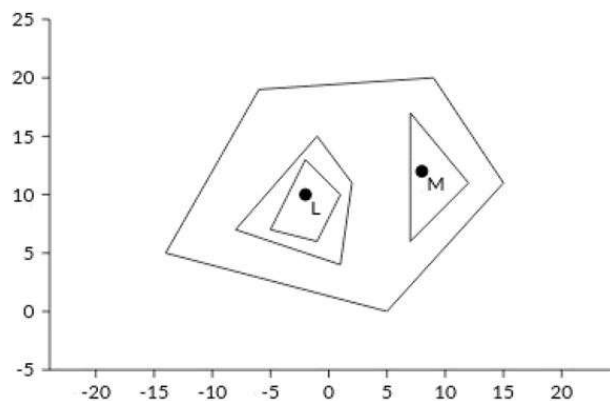


Figure 1: Illustration for valid placement in Example 1. Point M represents Michael and point L represents Lincoln

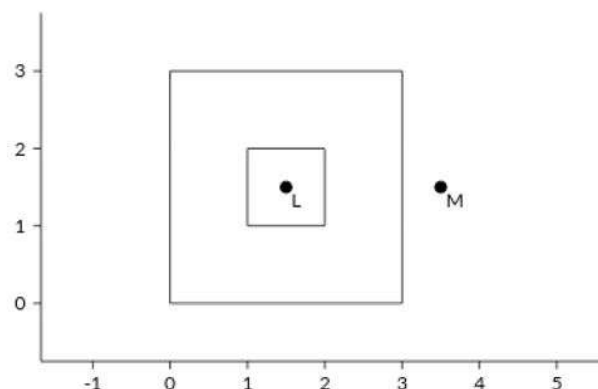


Figure 2: Illustration for valid placement in Example 2

Input

The first line of the input contains one integer N ($1 \leq N \leq 2 \times 10^5$), the number of convex polygons. This line is followed by the descriptions of each polygon. The i -th description starts with an integer k_i ($3 \leq k_i \leq 6 \times 10^5$) followed by k_i lines, each line contains a point (x_j, y_j) ($-10^9 \leq x_j, y_j \leq 10^9$).

The points in the order they are given form a convex polygon in counter-clockwise order and no three consecutive points from the polygon are collinear. No two edges from different polygons intersect. The total number of edges does not exceed 6×10^5 , that is $\sum_{i=1}^N k_i \leq 6 \times 10^5$.

Output

Print one integer, the minimum number of walls that need to be climbed by Michael to rescue his brother, supposing that Brad assigned the brothers places so that such number of walls is maximum.

Input example 1 4 4 1 10 -2 13 -5 7 -1 6 5 15 11 9 20 -6 19 -14 5 5 0 4 -1 15 -8 7 1 4 2 11 3 7 17 7 6 12 11	Output example 1 6
Input example 2 2 4 0 0 3 0 3 3 0 3 4 1 1 2 1 2 2 1 2	Output example 2 4

Problema F

Fugarse de la prisión

Michael y su hermano Lincoln han sido injustamente encarcelados en la misma prisión, pero Michael tiene un plan para rescatar a su hermano. La prisión se puede ver como un conjunto de polígonos convexos en el plano, en el cual las aristas de los polígonos son paredes. Las paredes de polígonos distintos no se intersectan, pero los polígonos pueden ser anidados, es decir, estar uno dentro de otro. Michael y Lincoln pueden ser vistos como dos puntos en el plano. Para Michael, el camino de rescate consistirá en alcanzar a su hermano y luego ambos necesitan escapar de la prisión.

Caminar para ellos no es un problema, pero trepar paredes es peligroso y difícil, por lo que Michael intentará minimizar la cantidad total de paredes trepadas por él. Michael primero necesita trepar algunas paredes para alcanzar a su hermano si no están en la misma área, y luego trepar más paredes para salir de la prisión. Salir de la prisión quiere decir no estar dentro de ningunas paredes, o también puede verse como alcanzar un punto muy lejano, digamos $(10^{20}, 10^{20})$. Brad está a cargo del posicionamiento de los prisioneros y es consciente del plan de Michael, por lo que ubicará a los prisioneros en dos puntos distintos en el plano, no contenidos por ningún segmento, de modo que la menor cantidad de paredes que deben ser trepadas por Michael sea lo mayor posible. ¿Cuál es la mínima cantidad de paredes que serán trepadas por Michael si Brad coloca a los hermanos de manera óptima?

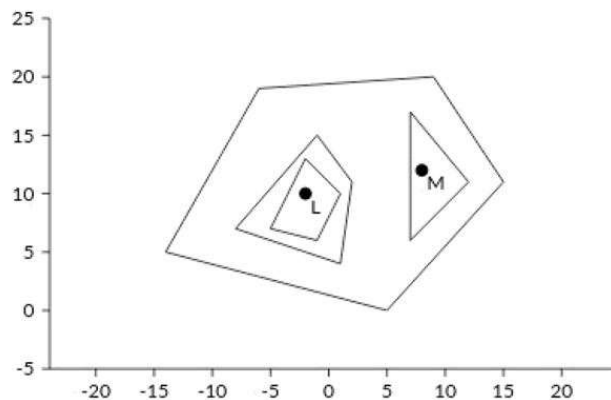


Figura 1: Ilustración de un posicionamiento válido para el ejemplo 1. El punto M representa a Michael y el punto L a Lincoln

Entrada

La primera línea de la entrada contiene un entero N ($1 \leq N \leq 2 \times 10^5$), la cantidad de polígonos convexos. Esta línea es seguida por las descripciones de cada polígono. La i -ésima descripción empieza con un entero k_i ($3 \leq k_i \leq 6 \times 10^5$) seguido de k_i líneas, cada una de las cuales contiene un punto (x_j, y_j) ($-10^9 \leq x_j, y_j \leq 10^9$).

Los puntos forman un polígono convexo y son dados en orden contrario a las agujas del reloj. No hay tres puntos consecutivos del polígono que sean colineales. No hay dos aristas de distintos polígonos que se intersequen. La cantidad total de aristas no excede 6×10^5 , es decir $\sum_{i=1}^N k_i \leq 6 \times 10^5$.

Salida

Imprime un entero, la menor cantidad de paredes que deben ser trepadas por Michael para rescatar a su hermano, en caso que Brad asignara a los hermanos posiciones de modo que tal cantidad de paredes sea máxima.

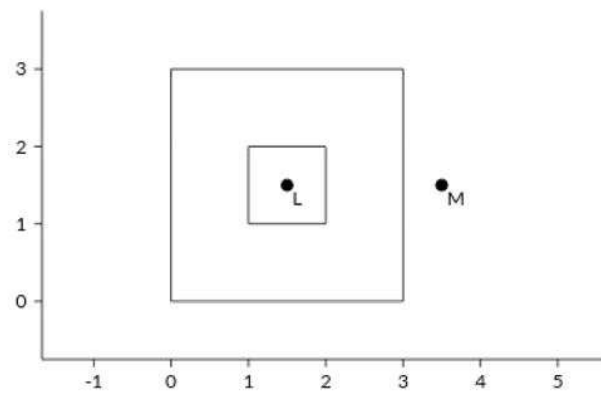


Figura 2: Ilustración de un posicionamiento válido para el ejemplo 2

Ejemplo de entrada 1	Ejemplo de salida 1
4 4 1 10 -2 13 -5 7 -1 6 5 15 11 9 20 -6 19 -14 5 5 0 4 -1 15 -8 7 1 4 2 11 3 7 17 7 6 12 11	6

Ejemplo de entrada 2	Ejemplo de salida 2
<pre>2 4 0 0 3 0 3 3 0 3 4 1 1 2 1 2 2 1 2</pre>	<pre>4</pre>

Problema G

Garantindo o Treino

Juan decidiu começar a se exercitar e está começando a preparar uma sessão de treinamento.

Ele sabe que em alguns dias pode não querer fazer todos os exercícios de sua sessão. Ele então decidiu criar algumas regras para evitar pular a sessão inteira e acabar não se exercitando, mas de forma que ainda possa evitar alguns exercícios de vez em quando.

As regras são:

- Haverá dois tipos de exercícios: A e B .
- Após terminar um exercício do tipo B ele fará o próximo exercício, se houver algum. Caso contrário, a sessão termina.
- Após terminar um exercício do tipo A , há duas possibilidades: ele pode iniciar o exercício seguinte ou pulá-lo, executando o exercício imediatamente posterior.
- O último exercício da sessão será sempre do tipo B .

Assim, podem existir diferentes maneiras de se completar a sessão de exercícios. Por exemplo, se os tipos dos exercícios são $BAAB$, existem 3 maneiras de se completar a sessão: fazendo todos os exercícios, pulando o terceiro ou pulando o último.

Juan quer preparar sua sessão de exercícios de maneira que existam exatamente N maneiras diferentes de completá-la. Você pode ajudá-lo?

Entrada

A entrada é composta por uma única linha contendo um inteiro N ($2 \leq N \leq 10^{15}$), representando o número de maneiras que a sessão de exercícios pode ser completada.

Saída

Imprima uma linha, formada pelos caracteres ‘A’ e ‘B’, representando os tipos dos exercícios na sessão. Se houver múltiplas respostas, imprima aquela que é lexicograficamente menor. Se não houver sessões válidas, imprima uma linha contendo a string “IMPOSSIBLE” (sem aspas).

Exemplo de entrada 1 2	Exemplo de saída 1 AB
Exemplo de entrada 2 4	Exemplo de saída 2 ABAB
Exemplo de entrada 3 7	Exemplo de saída 3 IMPOSSIBLE

Problem G

Getting in Shape

Juan decided to start working out and is willing to prepare a workout session.

He knows that some days he might not want to do all exercises from his workout session. He decided on some rules to avoid skipping the whole session and not exercising at all, while still allowing him to optionally skip some exercises.

The rules are:

- There will be only two types of exercises: A and B .
- After finishing an exercise of type B he moves to the next exercise, if there is one. Otherwise, the workout session ends.
- After finishing an exercise of type A there are two possibilities: he can move to the next exercise, or he can skip the next exercise and move to the one after that.
- The last exercise in a workout session must always be of type B .

Therefore, there might be different ways in which the workout session can be completed. For example, if the types of exercises in a workout session are **BAAB**, there are 3 ways in which the session can be completed: by doing all exercises, by skipping the 3rd one or by skipping the last one.

Juan wants to prepare his workout session in such a way that there are exactly N different ways in which the workout session can be completed. Can you help him?

Input

One positive integer N ($2 \leq N \leq 10^{15}$), representing the number of ways in which the workout session can be completed.

Output

Output a line containing a string, formed only with characters ‘A’ and ‘B’, representing the types of the exercises in the workout session. If there are multiple valid answers, output the lexicographically smallest answer. If there is no valid workout sessions, output a line containing the string “IMPOSSIBLE” (without quotes).

Input example 1 2	Output example 1 AB
Input example 2 4	Output example 2 ABAB
Input example 3 7	Output example 3 IMPOSSIBLE

Problema G

Garantizando el entrenamiento

Juan decidió comenzar a hacer ejercicio y se está preparando para su rutina de entrenamiento.

Sabe que algunos días podría no querer hacer todos los ejercicios de su rutina. Así que ha establecido algunas reglas para evitar saltarse todos los ejercicios de su rutina y no entrenar en absoluto, pero permitiéndose omitir opcionalmente algunos de los ejercicios.

Las reglas son:

- Solo habrá dos tipos de ejercicios: A , y B .
- Después de finalizar un ejercicio de tipo B , Juan pasará al siguiente ejercicio en caso de que haya alguno. De otro modo, la rutina habrá terminado.
- Después de finalizar un ejercicio de tipo A , hay dos posibilidades: Juan puede pasar al siguiente ejercicio, o, puede saltarse el siguiente ejercicio y pasar al siguiente después de ese.
- El último ejercicio de la rutina siempre debe ser del tipo B .

Entonces, puede haber diferentes maneras en las que Juan puede completar su rutina. Por ejemplo, si los tipos de los ejercicios en una rutina son: **BAAB**, hay 3 maneras en las que la rutina puede ser completada: Hacer todos los ejercicios, saltarse el tercer ejercicio, o saltarse el último ejercicio.

Juan quiere preparar su rutina de entrenamiento de tal manera que haya exactamente N diferentes formas en las que su rutina pueda ser completada. ¿Puedes ayudarlo?

Entrada

Un entero positivo N ($2 \leq N \leq 10^{15}$), que indica la cantidad de formas diferentes en las que la rutina de entrenamiento debe ser completada.

Salida

Imprima una línea que contenga una cadena, formada únicamente por los caracteres ‘A’, y ‘B’, que indiquen los tipos de ejercicios en la rutina de entrenamiento. En caso de que haya más de una respuesta válida, imprima la menor lexicográficamente. En caso de que no haya una rutina de entrenamiento válida, imprima una línea que contenga la cadena “IMPOSSIBLE” (sin las comillas).

Ejemplo de entrada 1 2	Ejemplo de salida 1 AB
Ejemplo de entrada 2 4	Ejemplo de salida 2 ABAB
Ejemplo de entrada 3 7	Ejemplo de salida 3 IMPOSSIBLE

Problema H

Haja Ordenação

Um amigo seu inventou um jogo e quer saber se você consegue resolvê-lo ou se ele é impossível.

Ele montou uma sequência de N blocos. Cada bloco tem um número gravado e uma cor. Todos os números são números distintos entre 1 e N , e blocos diferentes podem ter a mesma cor.

O jogo funciona da seguinte maneira: você pode jogar quantos turnos você quiser. Em um turno, você escolhe dois blocos diferentes que têm a mesma cor e os troca de posição.

Você deve dizer se é possível fazer com que a sequência inteira fique em ordem crescente ou não.

Entrada

A primeira linha contém dois inteiros N e K ($1 \leq N \leq 10^5$, $1 \leq K \leq N$), representando o número de blocos na sequência e o número de cores diferentes, respectivamente.

Cada uma das N linhas seguintes contém dois inteiros n_i e c_i ($1 \leq n_i \leq N$, $1 \leq c_i \leq K$), representando o número e a cor do i -ésimo bloco, respectivamente.

Saída

Imprima uma linha contendo um caractere. Se a sequência puder ser ordenada em ordem crescente, imprima a letra maiúscula 'Y'; caso contrário, imprima a letra maiúscula 'N'.

Exemplo de entrada 1 4 2 3 1 4 2 1 1 2 2	Exemplo de saída 1 Y
Exemplo de entrada 2 4 2 2 1 4 2 1 1 3 2	Exemplo de saída 2 N
Exemplo de entrada 3 3 1 1 1 2 1 3 1	Exemplo de saída 3 Y

Problem H

Handling the Blocks

A friend of yours invented a game and wants to know if you can solve it or if it's impossible.

He assembled a sequence of N blocks. Each block has a number engraved on it and some color. All numbers are distinct and between 1 and N , and different blocks can be of the same color.

The game works as follows: you can play as many turns as you want. In one turn, you choose two different blocks that share the same color and swap them.

You have to tell whether it is possible to get the entire sequence to be sorted into ascending order by numbers engraved on the blocks.

Input

The first line contains two integers N and K ($1 \leq N \leq 10^5$, $1 \leq K \leq N$), representing the number of blocks in the sequence and the number of different colors, respectively.

Each of the next N lines contains two integers n_i and c_i ($1 \leq n_i \leq N$, $1 \leq c_i \leq K$), representing the number and color of the i -th block, respectively.

Output

Output one line containing one character. If the sequence can be arranged in ascending order, write the upper case letter 'Y'; otherwise write the uppercase letter 'N'.

Input example 1 4 2 3 1 4 2 1 1 2 2	Output example 1 Y
Input example 2 4 2 2 1 4 2 1 1 3 2	Output example 2 N
Input example 3 3 1 1 1 2 1 3 1	Output example 3 Y

Problema H

Habr  ordenamiento

Un amigo tuyo ha inventado un juego y quiere saber si eres capaz de resolverlo.

Tu amigo ha construido una secuencia de N bloques. Cada bloque tiene un n mero escrito, y es de un color. Todos los n meros son distintos, entre 1 y N , pero diferentes bloques pueden tener el mismo color.

En el juego se pueden llevar a cabo tantos turnos como quieras. En cada turno, debes elegir dos bloques diferentes que tengan el mismo color e intercambiarlos.

Debes decir si es posible hacer que toda la secuencia est  en orden ascendente de acuerdo a los n meros escritos en los bloques.

Entrada

La primera l nea contiene dos enteros N y K ($1 \leq N \leq 10^5$, $1 \leq K \leq N$), que representan respectivamente, el n mero de bloques en la secuencia y el n mero de colores diferentes.

Cada una de las siguientes N l neas contiene dos n meros enteros n_i , y c_i ($1 \leq n_i \leq N$, $1 \leq c_i \leq K$), que representan respectivamente, el n mero que tiene escrito, y el color del i - simo bloque.

Salida

Debes imprimir una l nea que contenga solo una letra. Si la secuencia puede ser acomodada en orden ascendente, imprime la letra may scula ‘Y’; En caso contrario, imprime la letra may scula ‘N’.

Ejemplo de entrada 1 4 2 3 1 4 2 1 1 2 2	Ejemplo de salida 1 Y
Ejemplo de entrada 2 4 2 2 1 4 2 1 1 3 2	Ejemplo de salida 2 N
Ejemplo de entrada 3 3 1 1 1 2 1 3 1	Ejemplo de salida 3 Y

Problema I

Invertendo Ferrovias

O governo de Nlogônia está incomodado com a falta de eficiência de seu sistema ferroviário. Todo par de cidades possui uma única ferrovia que as liga, porém, devido a problemas monetários, algumas delas estão inativas.

Uma configuração ideal da malha ferroviária é tal que, para qualquer par de cidades, existe um único caminho ligando essas duas cidades usando somente ferrovias ativas.

O governador de Nlogônia te contratou para transformar seu conjunto de ferrovias em uma configuração ideal. Infelizmente, você não fala Nlogoniano e nem tem controle sobre as ferrovias: somente os líderes de cada cidade, que só falam Nlogoniano, podem ativar ou desativar as ferrovias.

Nlogoniano possui algumas expressões extremamente específicas. Para tentar te ajudar, e te desafiar ao mesmo tempo, um amigo te ensinou uma frase que você pode tentar usar: “`lupDujHomwIj luteb gharghmey`”. Ele disse que, ao falar isso para o líder de uma cidade, tal líder ativará todas as ferrovias adjacentes a essa cidade que estavam previamente desativadas e desativará todas as ferrovias adjacentes a essa cidade que estavam previamente ativas. Em outras palavras, o status de “ativação” de todas as ferrovias adjacentes à cidade em questão será trocado.

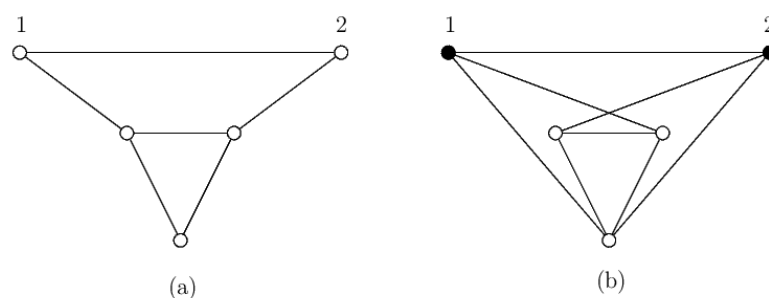


Figura 3: Resultado de “`lupDujHomwIj luteb gharghmey`” cidades 1 e 2

Sabendo essa frase, você pode ligar para alguns líderes e fazê-los “`lupDujHomwIj luteb gharghmey`” as ferrovias de suas cidades.

Seu amigo duvida que você conseguirá transformar o sistema de ferrovias em uma configuração ideal apenas utilizando essa frase. Você realmente quer provar que ele está errado. Você não só vai achar uma solução, mas, mais do que isso, vai lhe dizer de quantas maneiras diferentes pode completar o objetivo. Mais precisamente, digamos que um conjunto de líderes é *bom* se contatando precisamente cada um dos líderes desse conjunto uma única vez, a configuração ideal é obtida. Você vai dizer ao seu amigo quantos conjuntos bons distintos existem.

Dois conjuntos de líderes são considerados distintos se existe pelo menos um líder contido em um conjunto que não está contido no outro.

Como esse número pode ser muito grande, você deverá informá-lo módulo $10^9 + 7$.

Entrada

A primeira linha contém dois inteiros, N e M ($1 \leq N \leq 100$, $0 \leq M \leq N \times (N - 1)/2$), representando o número de cidades e o número de ferrovias inicialmente ativas, respectivamente.

Cada uma das próximas M linhas contém dois inteiros, u e v ($1 \leq u < v \leq N$), representando a existência de uma ferrovia inicialmente ativa entre as cidades u e v . É garantido que não existem duas dessas linhas iguais.

Saída

Imprima uma linha contendo a quantidade de conjuntos de líderes que tornam a configuração ideal módulo $10^9 + 7$.

Exemplo de entrada 1 5 6 1 2 1 3 2 4 3 4 3 5 4 5	Exemplo de saída 1 8
Exemplo de entrada 2 3 2 1 2 2 3	Exemplo de saída 2 6
Exemplo de entrada 3 4 4 1 2 2 3 3 4 1 3	Exemplo de saída 3 4
Exemplo de entrada 4 3 1 1 2	Exemplo de saída 4 0
Exemplo de entrada 5 2 0	Exemplo de saída 5 2

Exemplo de entrada 6	Exemplo de saída 6
10 15 1 6 1 2 1 5 6 7 6 10 2 3 2 9 7 3 7 8 3 4 8 9 8 5 4 5 4 10 9 10	0

Problem I

Inverting Everything

The government in Nlogonia is bothered by the lack of efficiency of their railroad system. Each pair of cities is connected by a single railroad, but due to budgeting issues, some of these are inactive.

An ideal railroad configuration is one such that for every pair of cities there is exactly one path that connects those cities using only active railroads.

The Nlogonian governor hired you to transform his set of railroads in an ideal configuration. You, unfortunately, don't speak Nlogonian nor can change the activeness of the railroads: only the leader of each city, all of which only speak Nlogonian, can activate or deactivate them.

Nlogonian has some really specific phrases, so you hope you can count on that. You have a friend that speaks Nlogonian, and to help you (and also challenge you a little bit), he taught you a phrase that you can try using: “lupDujHomwIj luteb gharghmey”. He said that if you say that to a city's leader, then the leader will activate all railroads that are connected to their city and that were previously inactive, and deactivate all railroads that are connected to their city and that were previously active. In other words, the “activation” status of all railroads that connect to that leader's city will be flipped.

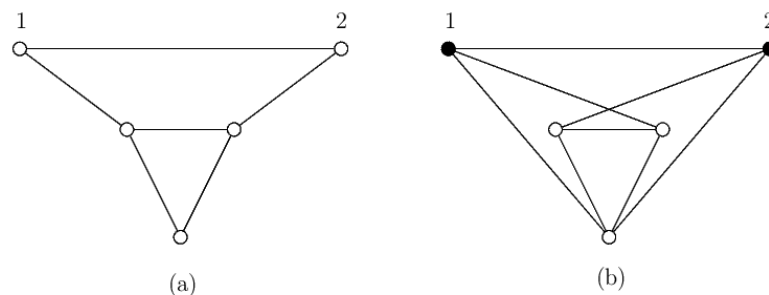


Figure 3: Result of “lupDujHomwIj luteb gharghmey” cities 1 and 2

Knowing this phrase, you can call some leaders and make them “lupDujHomwIj luteb gharghmey” their cities' railroads.

Your friends doubt that you can make the railroad system ideal by only using this phrase. You really want to prove them wrong. You said you will not only find a solution but tell in how many different ways you can achieve that. More precisely, we say a set of leaders is *good* if by contacting them and only them once, an ideal configuration is achieved. You want to tell your friend how many distinct good sets of leaders there are.

Two sets of leaders are distinct if there is at least one leader that is in one of the sets but not the other.

Since this number may be large, you must output it modulo $10^9 + 7$.

Input

The first line contains two integers, N and M ($1 \leq N \leq 100$, $0 \leq M \leq N \times (N-1)/2$), representing the number of cities and the number of railroads that are initially active.

Each of the next M lines contains two integers, u and v ($1 \leq u < v \leq N$), representing the existence of an initially active railroad between cities u and v . No pair of cities will be listed twice.

Output

Print one line consisting of the amount of *good* sets given the initial railroad configuration, modulo $10^9 + 7$.

Input example 1 5 6 1 2 1 3 2 4 3 4 3 5 4 5	Output example 1 8
Input example 2 3 2 1 2 2 3	Output example 2 6
Input example 3 4 4 1 2 2 3 3 4 1 3	Output example 3 4
Input example 4 3 1 1 2	Output example 4 0
Input example 5 2 0	Output example 5 2
Input example 6 10 15 1 6 1 2 1 5 6 7 6 10 2 3 2 9 7 3 7 8 3 4 8 9 8 5 4 5 4 10 9 10	Output example 6 0

Problema I

Invirtiendo todo

El gobierno de Nlogonia está molesto por la falta de eficiencia de su sistema ferroviario. Todos los pares de ciudades están conectados por una única vía de tren, pero debido a problemas de presupuesto, algunas de éstas están inactivas.

Una configuración ideal del tejido ferroviario es una tal que para todo par de ciudades existe exactamente un camino que conecta esas ciudades usando sólo vías activas.

El gobernador de Nlogonia te ha contratado para transformar su conjunto de vías en una configuración ideal. Desafortunadamente, no hablas nlogonio ni puedes activar o desactivar vías: Sólo los líderes de cada ciudad, todos los cuales solamente hablan nlogonio, pueden activarlas o desactivarlas.

El idioma nlogonio tiene algunas frases realmente específicas, así que esperas contar con ello. Tienes un amigo que habla nlogonio, y para ayudarte (y también desafiarte un poco), te ha enseñado una frase que puedes intentar usar: “`lupDujHomwIj luteb gharghmey`”. Te dijo que si dices eso al líder de una ciudad, entonces el líder activará todas las vías que están conectadas a su ciudad y que previamente estaban inactivas, y desactivará todas las vías que están conectadas a su ciudad y que previamente estaban activas. En otras palabras, los estados de “activación” de todas las vías que conectan a la ciudad de ese líder serán dados vuelta.

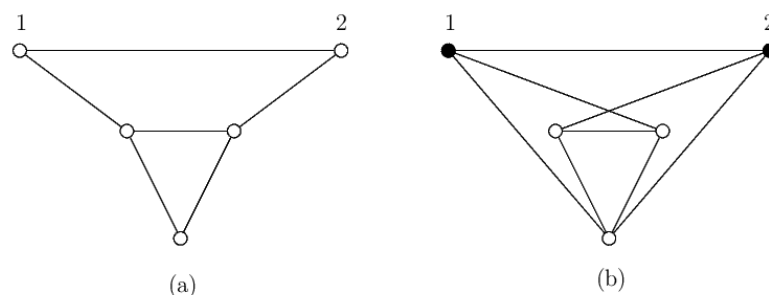


Figura 3: Resultado de “`lupDujHomwIj luteb gharghmey`” las ciudades 1 y 2

Conociendo esta frase, puedes llamar a algunos líderes y hacerles “`lupDujHomwIj luteb gharghmey`” las vías de sus ciudades.

Tu amigo duda que puedas hacer ideal el sistema ferroviario solamente usando esta frase. Realmente quieres probar que está equivocado. Tú le has dicho que no sólo encontrarás una solución, sino que también le dirás de cuántas formas diferentes lo puedes lograr. Más precisamente, decimos que un conjunto de líderes es *bueno* si contactando a ellos y sólo a ellos una vez se logra una configuración ideal. Quieres decirle a tu amigo cuántos conjuntos buenos de líderes hay.

Dos conjuntos de líderes son distintos si hay al menos un líder que está en uno de los conjuntos pero no en el otro.

Como este número puede ser muy grande, debes imprimir el resto de dividirlo por $10^9 + 7$.

Entrada

La primera línea contiene dos enteros, N y M ($1 \leq N \leq 100$, $0 \leq M \leq N \times (N - 1)/2$), que representan la cantidad de ciudades y la cantidad de vías que inicialmente están activas.

Cada una de las siguientes M líneas contiene dos enteros, u y v ($1 \leq u < v \leq N$), que representan la existencia de una vía inicialmente activa entre las ciudades u y v . Ningún par de ciudades será listado dos veces.

Salida

Imprime una línea conteniendo la cantidad de conjuntos *buenos* de líderes dada la configuración inicial del tejido ferroviario, módulo $10^9 + 7$.

Ejemplo de entrada 1 5 6 1 2 1 3 2 4 3 4 3 5 4 5	Ejemplo de salida 1 8
Ejemplo de entrada 2 3 2 1 2 2 3	Ejemplo de salida 2 6
Ejemplo de entrada 3 4 4 1 2 2 3 3 4 1 3	Ejemplo de salida 3 4
Ejemplo de entrada 4 3 1 1 2	Ejemplo de salida 4 0
Ejemplo de entrada 5 2 0	Ejemplo de salida 5 2

Ejemplo de entrada 6	Ejemplo de salida 6
<p>10 15 1 6 1 2 1 5 6 7 6 10 2 3 2 9 7 3 7 8 3 4 8 9 8 5 4 5 4 10 9 10</p>	<p>0</p>

Problema J

Jogo Duro

Futebol nem sempre foi o esporte mais popular das Américas. Historiadores encontraram registros de um antigo esporte que era jogado em muitas civilizações pelo continente. Por causa da falta da tradição de se falar sobre este esporte, o nome original é desconhecido, mas em tempos modernos ele é criativamente chamado de “butefol”.

Nós não sabemos muito sobre butefol, nem as regras básicas. Porém, arqueólogos encontraram muitas anotações feitas pelos técnicos enquanto eles montavam seus times, o que nos deu dicas sobre como os times eram formados. Estas anotações estão lotadas de números e cálculos. Os técnicos de butefol tentaram minuciosamente otimizar seus times ao atribuir os jogadores às melhores posições possíveis. Para facilitar esta tarefa, eles desenvolveram uma métrica para determinar a performance de cada arranjo.

Há M posições em um campo de butefol, que são distribuídas em uma linha. Um time de butefol é composto de N jogadores, cada um o qual é designado uma posição (todos os jogadores devem ser designados a uma posição, e cada posição pode ser ocupada por zero ou mais jogadores).

Naturalmente, os jogadores não são todos iguais: cada jogador pode ter performances diferentes quando joga em posições diferentes. Concretamente, para cada jogador i e cada posição j , há um inteiro positivo $P_{i,j}$ que representa a performance do jogador i quando jogando na posição j .

Para complicar as coisas ainda mais, os treinadores também consideravam o aspecto de interação dos jogadores. Há alguns pares de jogadores que são “melhores amigos”. Quando melhores amigos estão longes um do outro no campo, isto tem um impacto negativo na performance do time. Há um inteiro positivo C que representa a penalidade de performance quando melhores amigos estão longes um do outro.

Uma vez que os jogadores estejam distribuídos no campo, o valor da performance do time é calculado da seguinte maneira: primeiro, nós somamos a performance de cada jogador em sua determinada posição. Em seguida, para cada par de jogadores que são melhores amigos, nós subtraímos C multiplicado pela distância entre os dois jogadores, onde a distância entre dois jogadores é definida pela diferença (em valor absoluto) das posições onde os dois jogadores estão designados.

Nós gostaríamos de saber o quão bem os treinadores de butefol estavam formando seus times. Para isto, nós gostaríamos de saber qual é a maior performance possível de ser alcançada ao designar os jogadores de maneira ótima, dadas as performances dos jogadores em cada posição e os pares de jogadores que são melhores amigos.

Entrada

A primeira linha contém quatro inteiros N , M , K e C ($1 \leq N, M \leq 50$, $0 \leq K \leq 50$, $0 \leq C \leq 10^6$), representando a quantidade de jogadores, a quantidade de posições, a quantidade de pares de melhores amigos e a penalidade por colocar amigos longe uns dos outros.

Cada uma das N linhas seguintes contém M inteiros. O j -ésimo inteiro da i -ésima linha é $P_{i,j}$, representando a performance do jogador i se ele for designado na posição j ($0 \leq P_{i,j} \leq 10^6$).

Cada uma das K linhas seguintes contém 2 inteiros a_i e b_i ($1 \leq a_i < b_i \leq N$), que representa que os jogadores a_i e b_i são melhores amigos. Nenhum par de jogadores estará repetido nesta lista.

Saída

Imprima uma linha contendo um inteiro, representando o máximo de performance possível para o time.

Exemplo de entrada 1	Exemplo de saída 1
3 3 2 5 5 2 1 3 2 8 1 9 3 1 2 1 3	14

(Neste caso, a solução ótima é designar os jogadores 1 e 3 na posição 2, e o jogador 2 na posição 3, para que a soma da performance dos jogadores seja $2+8+9=19$, a penalidade seja 5 pelos jogadores 1 e 2 estarem distantes por 1 posição, e a penalidade seja 0 pelos jogadores 1 e 3 estarem na mesma posição).

Problem J

Just Bootfall

Football hasn't always been the most popular sport in the Americas. Historians have found records of an ancient sport that was played in many civilizations across the continent. Because of the lack of spoken tradition about it, the original name is unknown, but in modern times it has been very creatively named "bootfall".

We don't know a lot about bootfall, not even the basic rules. However, archeologists have found a lot of notes made by bootfall coaches when trying to assemble their teams, which give us some information about how teams were formed. These notes are filled with numbers and calculations. Bootfall coaches meticulously tried to optimize their team by assigning players to the best positions possible. To facilitate this task, they developed a metric for determining the performance of each arrangement.

There are M positions in a bootfall field, which are distributed in a line. A bootfall team consists of N players, each of which is assigned to some position (all players should be assigned to exactly one position, each position can be occupied by one or more players or can be left unoccupied).

Naturally, players are not equal to each other: Each player can have a different performance when playing in different positions in the field. Concretely, for each player i and each position j , there is a positive value $P_{i,j}$ which represents the performance of player i when playing in position j .

To complicate things further, coaches also consider the aspect of player interaction. Some pairs of players are "best friends". When best friends are far from each other in the field, that harms team performance. There's a positive value C which represents the performance penalty that is paid when moving best friends away from each other.

Once players are assigned to positions, the value of the team performance is calculated as follows: First, we add up the performances of the players when playing in their assigned position. Then, for each pair of players who are best friends, we subtract C times the distance between the two players, where the distance between two players is defined as the difference (in absolute value) between the positions to which the players were assigned.

We want to know how good bootfall coaches were at forming teams. In order to do that, we would like to know what is the maximum possible value of team performance achievable by arranging the players in the optimal positions, given the performances of the players in each position and the pairs of players who are best friends.

Input

The first line contains four integers N , M , K and C ($1 \leq N, M \leq 50$, $0 \leq K \leq 50$, $0 \leq C \leq 10^6$), representing the number of players, the number of positions, the number of pairs of close friends and the penalty for having close friends far from each other.

Each of the next N lines contains M integers. The j -th integer of the i -th line is $P_{i,j}$, representing the performance of player i if playing in position j ($0 \leq P_{i,j} \leq 10^6$).

Each of the next K lines contains 2 integers a_i and b_i ($1 \leq a_i < b_i \leq N$), which represent that players a_i and b_i are close friends. No two pairs of players are repeated in this list.

Output

Output a line containing one integer, representing the maximum team performance possible.

Input example 1	Output example 1
3 3 2 5 5 2 1 3 2 8 1 9 3 1 2 1 3	14

(In this case, the optimal solution is to assign players 1 and 3 to position 2, and player 2 to position 3, so the sum of player performances is $2+8+9=19$, and we pay a penalty of 5 for players 1 and 2 being at distance 1, and penalty 0 for players 1 and 3 being at the same position).

Problema J

Jugando fútbol

El fútbol no siempre ha sido el deporte más popular en América. Los historiadores han encontrado registros de un deporte antiguo que era jugado en muchas civilizaciones a lo largo del continente. Debido a la falta de tradición oral acerca del mismo, su nombre original es desconocido, pero en tiempos modernos ha sido llamado muy creativamente “bútfol”.

No sabemos mucho acerca del bútfol, ni siquiera las reglas básicas. Sin embargo, los arqueólogos han encontrado un montón de notas hechas por entrenadores de bútfol al tratar de armar sus equipos, lo cual nos da algo de información acerca de cómo se formaban los mismos. Estas notas están repletas de números y cálculos. Los entrenadores de bútfol intentaban meticulosamente optimizar sus equipos, asignando jugadores a las mejores posiciones posibles. Para facilitar esta tarea, desarrollaron una métrica para determinar el rendimiento de cada asignación.

Hay M posiciones en un campo de bútfol, que están distribuidas en una línea. Un equipo de bútfol consta de N jugadores, cada uno de los cuales es asignado a alguna posición (todos los jugadores deben ser asignados a exactamente una posición, cada posición puede ser ocupada por uno o más jugadores o puede ser dejada sin ocupar).

Naturalmente, los jugadores no son iguales entre sí: Cada jugador puede tener diferente rendimiento cuando juega en diferentes posiciones en el campo. Concretamente, para cada jugador i y cada posición j , hay un valor positivo $P_{i,j}$ que representa el rendimiento del jugador i cuando juega en la posición j .

Para complicar las cosas aún más, los entrenadores también consideran el aspecto de la interacción entre jugadores. Hay algunos pares de jugadores que son “mejores amigos”. Cuando los mejores amigos están lejos uno de otro en el campo, eso tiene un impacto negativo en el rendimiento del equipo. Hay un valor positivo C que representa la penalidad de rendimiento que se paga por mover a los mejores amigos lejos uno de otro.

Una vez que los jugadores son asignados a sus posiciones, el valor del rendimiento del equipo se calcula de esta forma: Primero, sumamos los rendimientos de los jugadores cuando juegan en su posición asignada. Luego, por cada par de jugadores que son mejores amigos, sustraemos C por la distancia entre los dos jugadores, donde la distancia entre dos jugadores se define como la diferencia (en valor absoluto) entre las posiciones a las que los jugadores fueron asignados.

Queremos saber qué tan buenos eran los entrenadores de bútfol a la hora de armar equipos. Para hacer eso, nos gustaría saber cuál es el máximo valor posible para el rendimiento de un equipo que se puede conseguir asignando a los jugadores en las posiciones óptimas, dados los rendimientos de los jugadores en cada posición y los pares de jugadores que son mejores amigos.

Entrada

La primera línea contiene cuatro enteros N , M , K y C ($1 \leq N, M \leq 50$, $0 \leq K \leq 50$, $0 \leq C \leq 10^6$), que representan la cantidad de jugadores, la cantidad de posiciones, la cantidad de pares de mejores amigos y la penalidad por tener a los mejores amigos lejos uno de otro.

Cada una de las siguientes N líneas contiene M enteros. El j -ésimo entero de la i -ésima línea es $P_{i,j}$, que representa el rendimiento del jugador i si juega en la posición j ($0 \leq P_{i,j} \leq 10^6$).

Cada una de las siguientes K líneas contiene 2 enteros a_i y b_i ($1 \leq a_i < b_i \leq N$), que representan que los jugadores a_i y b_i son mejores amigos. No se repiten pares de jugadores en esta lista.

Salida

Imprime una línea conteniendo un entero, representando el máximo rendimiento posible para el equipo.

Ejemplo de entrada 1	Ejemplo de salida 1
3 3 2 5 5 2 1 3 2 8 1 9 3 1 2 1 3	14

(En este caso, la solución óptima es asignar los jugadores 1 y 3 a la posición 2, y el jugador 2 a la posición 3, entonces la suma de los rendimientos de los jugadores es $2+8+9=19$, y pagamos una penalidad de 5 por los jugadores 1 y 2 estando a distancia 1, y penalidad 0 por los jugadores 1 y 3 estando en la misma posición).

Problema K

Katmandu

Finalmente a pandemia está melhorando e você finalmente pode fazer a coisa com a qual esteve sonhando nos últimos anos: comer no seu restaurante favorito. Acontece que esse restaurante fica em Katmandu, mas tudo bem, é só você ir de avião.

O problema é que viajar de avião quase sempre te deixa muito cansado. Você se considera descansado se você consegue dormir por T minutos sem interrupção, ou seja, você nunca está acordado de um certo momento t até $t + T$. Além disso você dorme com muita facilidade: você consegue dormir no começo de qualquer minuto e acordar ao fim de qualquer minuto.

Claro que se você dormir demais você vai acabar perdendo todas as refeições que servem no vôo! Isso é completamente inaceitável: nenhuma oportunidade de comer de graça pode passar em branco.

Felizmente, a companhia aérea te mandou o cronograma completo do vôo: a duração do vôo, D minutos, o número de refeições que serão servidas, M , e o tempo a partir do início do vôo em que essas refeições serão servidas, y_i . Você precisa estar acordado no início do minuto em que a refeição será servida para comê-la, caso contrário você não será servido. Como você está sempre com fome, a refeição será devorada instantaneamente.

Agora você quer saber, para ter o vôo perfeito, você consegue ficar descansado e ainda assim comer todas as refeições durante o vôo?

Entrada

A primeira linha da entrada contém três inteiros, T , D , M ($1 \leq T, D \leq 10^5$, $0 \leq M \leq 1000$), que representam, respectivamente, o número de minutos consecutivos que você precisa dormir para ficar descansado, a duração do vôo e o número de refeições que serão servidas durante o vôo.

Cada uma das M linhas seguintes contém um inteiro y_i ($0 \leq y_i \leq D$). Esses inteiros representam os tempos nos quais cada refeição será servida, e são dados em ordem cronológica.

Saída

Imprima uma única linha contendo um único caractere. Se você consegue descansar durante o vôo e ainda assim comer todas as refeições, imprima 'Y'; caso contrário, imprima 'N'.

Exemplo de entrada 1 3 10 3 2 4 7	Exemplo de saída 1 Y
Exemplo de entrada 2 4 10 3 2 4 7	Exemplo de saída 2 N
Exemplo de entrada 3 5 5 0	Exemplo de saída 3 Y

Exemplo de entrada 4 4 8 2 5 7	Exemplo de saída 4 Y
Exemplo de entrada 5 4 8 2 3 4	Exemplo de saída 5 Y

Problem K

Kathmandu

The pandemic is getting better and you can finally do the thing you’ve been dreaming of for the past few years: eat at your favorite restaurant! The restaurant happens to be in Kathmandu, but that’s fine, you can always take a plane.

The problem is that planes almost always leave you restless. You consider yourself properly rested if you can sleep for T uninterrupted minutes, which means you are never awake from a certain moment t to $t + T$. Also, you’re a very easy sleeper: you can fall asleep at the start of any minute and wake up at the end of any minute.

Of course, if you sleep too much you will miss all the airplane meals! That is unacceptable, as no opportunity for free food should go to waste.

Luckily, the airplane company sent you the whole flight schedule: the duration of the flight, D minutes, the number of meals that are going to be served, M , and the exact time they will serve the meals, y_i . You need to be awake at the time the meal is being served to be able to eat it, otherwise, the steward will not serve you. Since you’re always hungry, you will devour the meal instantly.

Now you are wondering, for the optimal plane traveling experience, can you get properly rested and still eat all meals during the flight?

Input

The first line of input contains three integers, T , D , M ($1 \leq T, D \leq 10^5$, $0 \leq M \leq 1000$), representing, respectively, the number of minutes you need to sleep without interruption to be properly rested, the duration of the flight and the number of meals that are going to be served during the flight.

Each of the next M lines contains an integer y_i ($0 \leq y_i \leq D$). These integers represent the times at which each meal is going to be served, and are given in chronological order.

Output

Output a line containing one character. If you can get properly rested and still eat all meals during the flight, write the upper case letter ‘Y’; otherwise write the uppercase letter ‘N’.

Input example 1 3 10 3 2 4 7	Output example 1 Y
Input example 2 4 10 3 2 4 7	Output example 2 N
Input example 3 5 5 0	Output example 3 Y

Input example 4 4 8 2 5 7	Output example 4 Y
Input example 5 4 8 2 3 4	Output example 5 Y

Problema K

Katmandú

Por fin la pandemia está mejorando y podrás hacer esa cosa con la que has estado soñando tanto durante los últimos años: comer en tu restaurante favorito. Resulta que este restaurante está en Katmandú, pero no hay problema, siempre puedes tomar un avión.

El problema es que los aviones casi siempre te dejan cansado. Consideras que tienes un descanso adecuado si puedes dormir por T minutos sin interrupción, esto es, no estás despierto en ningún momento entre t y $t + T$. Además, eres muy bueno durmiendo: te puedes quedar dormido al inicio de cualquier minuto y despertar al finalizar cualquier minuto.

Pero si duermes demasiado ¡Podrías perder las comidas que sirven en el vuelo! Eso sería inaceptable, ninguna oportunidad de tener comida gratis debe ser desperdiciada.

Afortunadamente, la aerolínea te ha enviado por anticipado el cronograma completo del viaje: la duración D del vuelo en minutos, el número M de comidas que se van a servir, y el momento y_i en el que cada una de estas comidas se va a servir. Para poder comer una de ellas, debes estar despierto en el momento en el que se sirve, de otro modo la aeromoza no te la servirá. Como siempre estás hambriento, devorarás la comida instantáneamente al servirse.

Ahora te preguntas, para tener la mejor experiencia en el avión, ¿Puedes tener un descanso adecuado, y además comer todas las comidas del vuelo?

Entrada

La primera línea de entrada contiene tres enteros, T , D , M ($1 \leq T, D \leq 10^5$, $0 \leq M \leq 1000$), que representan, respectivamente, el número de minutos que debes dormir sin interrupción para tener un descanso adecuado, la duración del vuelo, y el número de comidas que se van a servir durante el vuelo.

Cada una de las siguientes M líneas contiene un entero y_i ($0 \leq y_i \leq D$). Estos enteros representan los momentos en los que cada comida se va a servir, y están dados en orden cronológico.

Salida

Debes escribir una línea que contenga solo una letra. Si puedes tener un descanso adecuado y comer todas las comidas durante el vuelo, escribe la letra mayúscula ‘Y’; en caso contrario, escribe la letra mayúscula ‘N’.

Ejemplo de entrada 1 3 10 3 2 4 7	Ejemplo de salida 1 Y
Ejemplo de entrada 2 4 10 3 2 4 7	Ejemplo de salida 2 N
Ejemplo de entrada 3 5 5 0	Ejemplo de salida 3 Y

Ejemplo de entrada 4 4 8 2 5 7	Ejemplo de salida 4 Y
Ejemplo de entrada 5 4 8 2 3 4	Ejemplo de salida 5 Y

Problema L

Lembre sua Senha

Michael é o gerente de um escritório pouco conhecido, e dentro de sua sala existe um cofre com o dinheiro para pagar seus funcionários. Infelizmente, Michael esqueceu a senha do cofre, e agora é a responsabilidade de Dwight ajudar seu chefe.

A senha é uma sequência de N dígitos, contendo apenas zeros e uns, e Michael lembra do valor de apenas algumas posições da sequência, mas não da senha inteira. Michael também lembra de M intervalos da senha que são palíndromos – sua mente memoriza palíndromos, por algum motivo.

Um intervalo é um palíndromo se, e somente se, o primeiro e o último dígitos do intervalo são iguais, o segundo e o penúltimo dígitos são iguais, e assim por diante.

Agora Dwight quer saber o quão difícil vai ser recuperar a senha inteira. Você pode ajudar Dwight ao calcular o número de senhas possíveis que respeitam a memória de Michael.

Como a resposta pode ser muito grande, imprima-a módulo $10^9 + 7$.

Entrada

A primeira linha contém dois inteiros N e M ($1 \leq N \leq 3 \times 10^5$, $1 \leq M \leq 3 \times 10^5$), representando quantos dígitos a senha tem, e a quantidade de intervalos que Michael lembra que são palíndromos, respectivamente.

A segunda linha contém N caracteres s_i , representando qual dígito Michael lembra sobre cada posição da senha. Se s_i é '0' ou '1', então isto significa que o i -ésimo dígito da senha é 0 ou 1, respectivamente. Se s_i é '?', então isto significa que Michael não lembra qual é o i -ésimo dígito.

Cada uma das M linhas seguintes contém dois inteiros l_i e r_i ($1 \leq l_i \leq r_i \leq N$), que significa que o intervalo da senha iniciando da posição l_i até a posição r_i , inclusive, é um palíndromo.

Saída

Imprima o número de possíveis senhas que formam uma senha válida módulo $10^9 + 7$. Como a memória de Michael pode ser conflitante, caso não haja nenhuma senha que respeite suas memórias, imprima '0'.

Exemplo de entrada 1 5 2 1??0? 1 3 2 4	Exemplo de saída 1 2
Exemplo de entrada 2 3 2 ??? 1 1 1 3	Exemplo de saída 2 4
Exemplo de entrada 3 5 2 1???0 1 3 3 5	Exemplo de saída 3 0

Problem L

Listing Passwords

Michael is the manager of a barely known office and inside his room there is a locker with the money to pay the employees. Unfortunately, Michael forgot the password of the locker and it is now Dwight's responsibility to help his boss.

The password is a sequence of N digits, containing only zeroes and ones, and Michael remembers the value at some positions of the sequence, but not the whole password. Michael also remembers M intervals of the password that are palindromes – his mind memorizes palindromes, for some reason.

An interval is a palindrome if, and only if, the first and last digits of the interval are equal, the second and second to last digits are equal, and so on.

Now Dwight wants to know how hard it will be to recover the whole password. You can help Dwight by calculating the number of possible passwords that follow what Michael remembers.

Since the answer may be very large, output it modulo $10^9 + 7$.

Input

The first line contains the two integers N and M ($1 \leq N \leq 3 \times 10^5$, $1 \leq M \leq 3 \times 10^5$), representing how many digits the password has and the number of intervals Michael remembers as palindromes, respectively.

The second line contains N characters s_i , representing what digit Michael remembers about each position of the password. If s_i is '0' or '1', then the i -th digit of the password is 0 or 1, respectively. If s_i is '?', then Michael doesn't remember the i -th digit.

Each of the next M lines contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq N$), which means that the interval of the password from the digit at position l_i to the digit at position r_i , inclusive, is a palindrome.

Output

Output the number of possible passwords that form a valid password modulo $10^9 + 7$. Since some of Michael's memories may be conflicting, in case there are no passwords that follow all his memories, print '0'.

Input example 1 5 2 1??0? 1 3 2 4	Output example 1 2
Input example 2 3 2 ??? 1 1 1 3	Output example 2 4
Input example 3 5 2 1???0 1 3 3 5	Output example 3 0

Problema L

Listando contraseñas

Michael es el jefe de una oficina apenas conocida y dentro de su habitación hay una caja de seguridad con el dinero para pagarle a los empleados. Desafortunadamente, Michael olvidó la contraseña de la caja de seguridad y ahora es responsabilidad de Dwight ayudar a su jefe.

La contraseña es una secuencia de N dígitos, que contiene sólo ceros y unos, y Michael recuerda el valor en algunas posiciones de la secuencia, pero no la contraseña entera. Michael también recuerda M intervalos de la contraseña que son palíndromos – su mente memoriza palíndromos, por algún motivo.

Un intervalo es un palíndromo si, y sólo si, el primer y último dígitos son iguales, el segundo y el penúltimo son iguales, y así sucesivamente.

Ahora Dwight quiere saber qué tan difícil será recuperar la contraseña completa. Debes ayudar a Dwight, calculando el número de contraseñas posibles que son acordes a lo que Michael recuerda.

Como la respuesta puede ser muy grande, se debe imprimir el resto de dividirla por $10^9 + 7$.

Entrada

La primera línea contiene los dos enteros N y M ($1 \leq N \leq 3 \times 10^5$, $1 \leq M \leq 3 \times 10^5$), que representan cuántos dígitos tiene la contraseña y la cantidad de intervalos que Michael recuerda como palíndromos, respectivamente.

La segunda línea contiene N caracteres s_i , representando cuál dígito Michael recuerda para cada posición de la contraseña. Si s_i es '0' ó '1', entonces significa que el i -ésimo dígito de la contraseña es 0 ó 1 respectivamente. Si s_i es '?', entonces Michael no recuerda el i -ésimo dígito.

Cada una de las siguientes M líneas contiene dos enteros l_i y r_i ($1 \leq l_i \leq r_i \leq N$), lo cual significa que el intervalo de la contraseña desde el dígito en la posición l_i al dígito en la posición r_i , inclusive, es un palíndromo.

Salida

Imprime el resto de dividir el número de contraseñas posibles por $10^9 + 7$. Como algunos de los recuerdos de Michael pueden ser contradictorios, en caso de que no haya contraseñas que cumplan con todo lo que recuerda, imprime '0'.

Ejemplo de entrada 1 5 2 1??0? 1 3 2 4	Ejemplo de salida 1 2
Ejemplo de entrada 2 3 2 ??? 1 1 1 3	Ejemplo de salida 2 4
Ejemplo de entrada 3 5 2 1???0 1 3 3 5	Ejemplo de salida 3 0

Problema M

Monarquia em Vertigem

A ordem de sucessão da monarquia é um tópico complexo, pois pode envolver múltiplos fatores tal como descendência, gênero, legitimidade, e religião. Geralmente a Coroa é herdada pelo filho do soberano, ou pela linhagem lateral do soberano caso ele não tenha filhos. Não é muito óbvio, não é mesmo? E este é um dos motivos pelos quais, em todos os lugares do mundo, a monarquia está quase acabando.

A Nlogônia ainda é dominada pela monarquia, mas felizmente com regras de sucessão simples. Em resumo há dois aspectos para levar em consideração: “filhos vêm antes de irmãos” e “os mais velhos vêm antes dos mais novos”.

Os servos do reino mantêm uma linda e gigante tapeçaria onde a linhagem de Constante, o primeiro rei da Nlogônia, é desenhada em forma de uma árvore. Sempre que um novo membro da família nasce um novo ramo do pai para o filho é desenhado na tapeçaria. Este evento é tão importante que a lenda diz que quando um descendente de Constante tem um filho ele não vai morrer até que o nome do seu filho seja desenhado na tapeçaria. Quando alguém morre, uma cruz é desenhada perto do nome do falecido na tapeçaria. Quando o monarca morre, a tapeçaria é usada pelos servos para determinar quem deve ser o novo monarca. Para que isso seja feito, os servos analisam a árvore iniciando pelo Constante e atravessam os ramos de acordo com as regras descritas antes, “filhos vêm antes de irmãos” e “os mais velhos vêm antes dos mais novos”. Eles visitam cada nó da árvore iniciando pelo Constante, seguido pelo filho mais velho de Constante, seguido pelo filho mais velho daquele filho, e assim por diante, até que encontram alguma pessoa ainda viva, ou até que um membro da família não tenha filhos, e neste caso eles voltam para o pai daquela pessoa e movem para o seu segundo filho, repetindo este processo até que um novo monarca seja encontrado.

Depois de milhares de anos no poder, a linhagem de Constante é gigante. Manter a tapeçaria e, quando a hora chega, determinar quem é o novo monarca são processos demorados e os servos Nlogonianos decidiram que está na hora de modernizar. Eles querem escrever um programa que seja usado para manter a linhagem de Constante e que também possa definir quem é o novo monarca quando o monarca anterior tragicamente morre. Dada a importância desta tarefa, os servos da monarquia querem testar o programa garantindo que ele produza a saída correta para todos os eventos que aconteceram até agora. Só existe um problema: nenhum deles é bom em programação. Por isso eles querem a sua ajuda.

Mais tecnicamente, cada pessoa na linhagem de Constante vai ser representado por um identificador inteiro positivo único. Sempre que um novo filho nasce, ele é atribuído o próximo menor inteiro único. O identificador de Constante é o número 1, e inicialmente ele é a única pessoa viva. Você terá que processar vários eventos em ordem cronológica. Sempre que uma pessoa morrer, você deve ajudar os servos a descobrirem quem é o atual monarca. É garantido que sempre haverá alguém vivo para governar.

Entrada

A primeira linha contém um inteiro Q ($1 \leq Q \leq 10^5$), representando quantos eventos devem ser processados. As próximas Q linhas conterão dois inteiros t_i e x_i cada, representando o tipo e o argumento do i -ésimo evento. Se t_i é igual a 1, então isso significa que a pessoa com identificador x_i teve um filho. Se t_i é igual a 2, então isso significa que a pessoa com identificador x_i morreu.

Saída

Para cada evento em que uma pessoa morre, você deve imprimir uma linha com um inteiro, representando o identificador do monarca atual.

Exemplo de entrada 1 8 1 1 1 1 1 2 2 1 2 4 1 2 2 2 2 5	Exemplo de saída 1 2 2 5 3
Exemplo de entrada 2 4 1 1 1 1 2 2 2 1	Exemplo de saída 2 1 3

Problem M

Monarchy in Vertigo

Monarchy succession can be a tricky topic as it might take into account multiple factors such as descent, sex, legitimacy, and religion. Usually, the Crown is inherited by a sovereign's child or by a childless sovereign's nearest collateral line. Not really straightforward, right? And that's one of the reasons why, all over the world, monarchy is on the edge.

Nlogonia is still ruled by a monarchy though, luckily with simple succession rules. In general, there are only two aspects to take into account: “children comes before siblings” and “older comes before younger”.

The reign servants maintain a beautiful and enormous tapestry where the bloodline of Constant, Nlogonia's first ruler, is drawn in the shape of a tree. Whenever a new family member is born a new branch from the parent to the child is drawn in the tapestry. This is such an important event that legend says that when Constant's descendants have a child they will never die before watching their child's name added to the tapestry. When someone dies, a cross is drawn close to that person's name. Whenever the current monarch dies, the tapestry is used by the servants to determine who the next ruler should be. In order to determine who that person is, the servants start from Constant and traverse the tree according to the rules described earlier, “children comes before siblings” and “older comes before younger”. They descend the tree starting from Constant, followed by Constant's first child, followed by that child's first child, and so on, until reaching the first person alive or a family member that has no children left to follow, in which case they then go back to that person's parent and move into that parent's next child, repeating the process until the new monarch is determined.

After thousands of years in power Constant's bloodline is huge. To maintain the tapestry and, when the time comes, to determine who the next monarch should be are lengthy processes and Nlogonian servants decided it's time to modernize. They want to create a program to be used to maintain Constant's bloodline, which can also point who the next monarch should be after a ruler's tragic death. Given the importance of this task, the monarchy servants want to test the program by checking that it produces the correct output for all events that happened so far. There is one issue though, none of them is really good with programming and that's why they came after you for help.

More technically, each person in Constant's bloodline will be represented by a unique positive integer identifier. Whenever a new child is born, they take the next lowest unique identifier. Constant's identifier is equal to 1, and initially he's the only person alive. You will be given many events to process, in chronological order. Whenever someone dies, you should help the monarchy servants to figure out who the current monarch is. It's guaranteed that there will always be someone alive to rule.

Input

The first line contains an integer Q ($1 \leq Q \leq 10^5$), representing how many events should be processed. The following Q lines contain two integers t_i and x_i each, representing the i -th event type and argument. If t_i is equal to 1, then it means that person with identifier x_i had a new child. If t_i is equal to 2, then it means that person with identifier x_i died.

Output

For each event in which someone dies, you should print a line with an integer, representing the identifier of the current monarch.

Input example 1 8 1 1 1 1 1 2 2 1 2 4 1 2 2 2 2 5	Output example 1 2 2 5 3
Input example 2 4 1 1 1 1 2 2 2 1	Output example 2 1 3

Problema M

Monarquía vertiginosa

La sucesión monárquica es un asunto muy delicado, ya que se deben tener en cuenta múltiples factores como el parentesco, el sexo, la legitimidad y la religión. Habitualmente la corona es heredada por el hijo de un soberano, o por la línea colateral más cercana de un soberano sin hijos. No es algo sencillo ¿cierto? Y esa es una de las razones, por las cuales en todo el mundo la monarquía está al límite.

Nlogonia aún es gobernada por una monarquía, aunque por suerte posee reglas de sucesión simples. En general hay solamente dos aspectos a tener en cuenta: “Los hijos antes que los hermanos” y “los viejos antes que los jóvenes”.

Los sirvientes del reino mantienen un hermoso y enorme tapiz en donde se encuentra dibujado en forma de árbol todo el linaje de Constante, el primer gobernante de Nlogonia. Cada vez que nace un nuevo miembro de la familia, se dibuja una nueva rama desde el padre hasta el hijo en el tapiz. Este es un evento tan importante que, de acuerdo a la leyenda, cuando un descendiente de Constante tiene un hijo, nunca morirá antes de ver el nombre de su hijo agregado al tapiz.

Cuando alguien muere, se marca su nombre con una cruz en el tapiz. Cada vez que el monarca actual muere, el tapiz es utilizado por los sirvientes para determinar quién debe ser el próximo gobernante. Para determinarlo los sirvientes comienzan desde Constante y recorren el árbol de acuerdo a las reglas antes explicadas, “Los hijos antes que los hermanos” y “los viejos antes que los jóvenes”. Descienden por el árbol comenzando con Constante, luego el primer hijo de Constante, luego el primer hijo de ese hijo y así siguiendo, hasta llegar a la primera persona viva o a un miembro de la familia sin más hijos, en cuyo caso vuelven al padre de esa persona y pasan a su siguiente hijo, continuando con este proceso hasta determinar el nuevo monarca.

Luego de miles de años en el poder, el linaje de Constante es enorme. Mantener el tapiz y, cuando llega la hora, determinar quién debe ser el siguiente monarca son procesos largos y los sirvientes Nlogonios decidieron que ya es hora de modernizarse. Quieren crear un programa que se utilice para mantener el linaje de Constante, y que pueda además indicar quién debe ser el siguiente monarca tras la trágica muerte de un gobernante. Dada la importancia de esta tarea, los sirvientes de la monarquía desean probar el programa verificando que produzca la salida correcta para todos los eventos que ya ocurrieron. El único problema es que ninguno de ellos es bueno programando, y es por eso que han venido en busca de tu ayuda.

Formalmente, cada persona en el linaje de Constante se representa mediante un único identificador, que es un entero positivo. Cada vez que nace una nueva persona, su identificador es el siguiente número más chico disponible. El identificador de Constante es 1, e inicialmente él es la única persona viva. Recibirás muchos eventos para procesar, en orden cronológico. Cada vez que alguien muera, deberás ayudar a los sirvientes a descubrir quién es el monarca entonces. Se garantiza que siempre habrá alguien vivo para gobernar.

Entrada

La primera línea contiene un entero Q ($1 \leq Q \leq 10^5$), que indica la cantidad de eventos que se deben procesar.

Las siguientes Q líneas contienen dos enteros t_i y x_i cada una, que representan el tipo y el argumento del i -ésimo evento.

Si t_i es 1, el evento significa que la persona con identificador x_i tuvo un nuevo hijo. Si t_i es 2, el evento significa que la persona con identificador x_i murió.

Salida

Por cada evento en el que alguien muera, se debe escribir una línea con un entero, que indica el identificador del monarca luego de la muerte.

Ejemplo de entrada 1 8 1 1 1 1 1 2 2 1 2 4 1 2 2 2 2 5	Ejemplo de salida 1 2 2 5 3
Ejemplo de entrada 2 4 1 1 1 1 2 2 2 1	Ejemplo de salida 2 1 3

Problema N

Na Trave!

Vini é um pintor de carros muito dedicado. Desde que ele aprendeu como pintar carros, o seu sonho tem sido participar da Internacional Competição de Pintores de Carros (ICPC).

Todo ano a região de Vini tem uma competição local para classificar todos os times competitivos de pintores de carro da região. Pintores em times que se classificaram nas melhores x posições avançam para competir na ICPC. É uma competição muito emocionante com muitos competidores novos em todos os anos, até que a fumaça nociva das tintas eventualmente faz com que os competidores se aposentem permanentemente.

Por causa de variações de verba e também por restrições da ICPC, a quantidade x pode variar de ano pra ano, o que pode acabar causando desgosto de alguns dos competidores.

No último ano de Vini como competidor, o seu time estava a uma posição de se qualificar para a ICPC. Que azar! Para fazer com que o seu sentimento de “má sorte” ficasse ainda mais forte, no ano seguinte o time que obteve a mesma colocação se classificou para a ICPC! Apesar do sentimento, depois de falar com outros antigos competidores, ele notou que muitos deles já haviam se sentido azarados de uma forma ou de outra.

Antigos competidores geralmente seguem os resultados das competições regionais por alguns anos após se aposentarem. Portanto, um competidor não se sentiria azarado pelas mudanças em x que acontecessem muitos anos após se aposentar. Mais precisamente, cada antigo competidor participou da sua última competição no ano a_i , se posicionando na posição p_i e, após se aposentar, seguiu os resultados pelos f_i anos seguintes.

Um competidor que não se qualificou para a ICPC em sua última participação se sentiu azarado em todos os anos em que ele seguiu os resultados e nos quais ele teria se classificado se tivesse competido. Em outras palavras, para cada ano até f_i anos após se aposentar, se ele não se qualificou em sua última participação, ele se sentiu azarado se o número de times qualificados para a ICPC naquele ano foi ao menos p_i .

Dado o número de vagas por ano, e as informações sobre cada antigo competidor, nós gostaríamos de saber em quantos anos cada antigo competidor se sentiu azarado.

Entrada

A primeira linha contém dois inteiros Y e N ($1 \leq Y, N \leq 3 \times 10^5$), representando a quantidade de anos de competições e a quantidade de antigos competidores com quem Vini conversou, respectivamente. (Sim, pintar carros é uma tradição milenar, e bem popular!).

A próxima linha contém Y inteiros x_1, x_2, \dots, x_Y ($0 \leq x_i \leq 10^5$), representando quantas vagas para a ICPC a região teve em cada ano.

Cada uma das seguintes N linhas contém três inteiros a_i, p_i e f_i ($1 \leq a_i \leq Y, 1 \leq p_i \leq 10^5, 0 \leq f_i \leq Y - a_i$), representando o ano em que o i -ésimo antigo competidor teve sua última participação, a colocação do time do i -ésimo antigo competidor naquele ano, e por quantos anos o i -ésimo antigo competidor seguiu os resultados após se aposentar, respectivamente.

Saída

Imprima N linhas, onde a i -ésima linha deve conter um inteiro representando quantos anos o i -ésimo antigo competidor se sentiu azarado.

Exemplo de entrada 1 5 3 1 2 3 4 5 1 3 4 2 6 3 3 4 1	Exemplo de saída 1 3 0 1
Exemplo de entrada 2 4 1 8 8 8 8 1 7 3	Exemplo de saída 2 0

Problem N

No Luck

Vini is a very dedicated car painter. Ever since he learned how to paint cars, his dream was to participate in the International Competition for Painting Cars (ICPC for short).

Every year Vini's region has a local competition to rank all teams of competitive car painters from this region. Painters in teams that were ranked in the top x spots will then go on to compete in the ICPC. It is a very thrilling competition with lots of new competitors every year, until the noxious fumes from the car paint eventually leads them to permanently retire.

Because of the variance in the national car painting budget and ICPC constraints, this number x may vary between years, which may cause a lot of displeasure in some competitors.

On Vini's last year as a contestant, his team was a single position away from qualifying to the ICPC. How unlucky! To make his "bad luck" feeling worse, in the following year the team who got the same position did qualify! Despite this feeling, after talking to other former contestants, he noticed that many have had the same feeling of being unlucky in one way or another.

Former contestants usually follow the results of the regional competition for a few years after retiring. Hence, a contestant would not feel unlucky for changes in x that occur many years after they retire. More precisely, each former contestant had their last participation in the year a_i , being placed in the position p_i and, after retiring, followed results for the next f_i years.

A contestant that didn't qualify to the ICPC in their last participation has felt unlucky on every year that they followed the results in which they would qualify to the ICPC if they had competed in that year. In other words, for every year up to f_i years after the contestant retired, if they didn't qualify in their last participation, they felt unlucky if the number of teams that qualify for the ICPC in this year was at least p_i .

Given the number of slots per year and the information about the former contestants, we want to know how many years each former contestant felt unlucky.

Input

The first line contains two integers Y and N ($1 \leq Y, N \leq 3 \times 10^5$), representing the number of years of competition and the number of former contestants that Vini had talked to, respectively. (Yes, painting cars is a millenary tradition, also a very popular one!).

The next line contains Y integers x_1, x_2, \dots, x_Y ($0 \leq x_i \leq 10^5$), representing the how many slots to the ICPC for their region there were for each year.

Each of the next N lines contains three integers a_i, p_i and f_i ($1 \leq a_i \leq Y$, $1 \leq p_i \leq 10^5$, $0 \leq f_i \leq Y - a_i$), representing the year that the i -th former competitor had their last participation, the position the i -th former competitor's team ranked and for how many years after competing the i -th former competitor followed the results after retiring, respectively.

Output

Output N lines, the i -th line should contain how many years the i -th former competitor felt unlucky.

Input example 1	Output example 1
5 3	3
1 2 3 4 5	0
1 3 4	1
2 6 3	
3 4 1	

Input example 2	Output example 2
4 1 8 8 8 8 1 7 3	0

Problema N

No tienen suerte

Vini es un pintor de carros muy dedicado. Desde que aprendió a pintar carros su sueño ha sido participar en la Competencia Internacional de Pintar Carros (ICPC por sus siglas en inglés).

Cada año la región de Vini tiene una competencia local para clasificar a los equipos de pintores de carros de esta región. Los pintores en los equipos que fueron clasificados en los mejores x lugares irán a competir al ICPC. Es una competencia muy emocionante, con muchos nuevos competidores cada año, hasta que los humos nocivos de la pintura de carros eventualmente los lleva a retirarse permanentemente.

Debido a la variación en el presupuesto nacional de la pintura de carros y las restricciones del ICPC, el número x puede variar entre años, lo que puede causarle disgusto a algunos competidores.

En el último año de Vini como competidor, su equipo estuvo a solo un lugar de clasificar al ICPC. ¡Qué mala suerte! Para hacer que su “mala suerte” se sienta aún peor, ¡en el siguiente año el equipo que obtuvo la misma posición clasificó! A pesar de este sentimiento, después de hablar con otros ex competidores, se dio cuenta de que muchos habían tenido el mismo sentimiento de tener mala suerte de una u otra forma.

Los ex competidores normalmente siguen los resultados de la competencia regional por algunos años después de retirarse. Entonces, un competidor podría no sentirse con mala suerte por los cambios que pudieran ocurrir en x algunos años después de retirarse. Siendo más precisos, cada ex competidor tuvo su última participación en el año a_i , obteniendo la posición p_i , y, después de retirarse, siguió los resultados por los siguientes f_i años.

Un competidor que no clasificó al ICPC en su último año de participación se ha sentido con mala suerte cada año de los que siguió los resultados en los que hubiera clasificado al ICPC si hubiera competido. En otras palabras, para cada año hasta f_i años después de que se retiró, si no clasificó en su última participación, se sintió con mala suerte si el número de equipos que clasificaron al ICPC en ese año fue al menos p_i .

Dados el número de clasificados por año y la información sobre cada ex competidor, queremos saber para cada ex competidor, ¿Cuántos años se sintió con mala suerte?.

Entrada

La primera línea de entrada contiene dos enteros Y y N ($1 \leq Y, N \leq 3 \times 10^5$), que representan respectivamente, el número de años de competencias y el número de ex competidores con los que Vini ha hablado. (Sí, ¡pintar carros es una tradición milenaria, y muy popular!).

La siguiente línea contiene Y enteros x_1, x_2, \dots, x_Y ($0 \leq x_i \leq 10^5$), que indican cuántos equipos clasificados al ICPC hubo en la región en cada año.

Cada una de las siguientes N líneas contiene tres números enteros a_i, p_i , y f_i ($1 \leq a_i \leq Y$, $1 \leq p_i \leq 10^5$, $0 \leq f_i \leq Y - a_i$), que representan respectivamente, el año en el que el i -ésimo ex competidor tuvo su última participación, la posición que obtuvo el i -ésimo competidor en su último año de participación, y por cuántos años el i -ésimo ex competidor siguió los resultados después de retirarse.

Salida

Imprima N líneas, la i -ésima línea debe contener cuántos años el i -ésimo ex competidor se sintió con mala suerte.

Ejemplo de entrada 1 5 3 1 2 3 4 5 1 3 4 2 6 3 3 4 1	Ejemplo de salida 1 3 0 1
Ejemplo de entrada 2 4 1 8 8 8 8 1 7 3	Ejemplo de salida 2 0