

A. Mike and Cellphone

Time limit: 1s

Memory limit: 256 MB

While swimming at the beach, Mike has accidentally dropped his cellphone into the water. There was no worry as he bought a cheap replacement phone with an old-fashioned keyboard. The keyboard has only ten digital equal-sized keys, located in the following way:

1	2	3
4	5	6
7	8	9
0		

Together with his old phone, he lost all his contacts and now he can only remember the way his fingers moved when he put some number in. One can formally consider *finger movements* as a sequence of vectors connecting centers of keys pressed consecutively to put in a number. For example, the finger movements for number "586" are the same as finger movements for number "253":



Mike has already put in a number by his "finger memory" and started calling it, so he is now worrying, can he be sure that he is calling the correct number? In other words, is there any other number, that has the same finger movements?

Input

The first line of the input contains the only integer n ($1 \leq n \leq 9$) — the number of digits in the phone number that Mike put in.

The second line contains the string consisting of n digits (characters from '0' to '9') representing the number that Mike put in.

Output

If there is no other phone number with the same finger movements and Mike can be sure he is calling the correct number, print "YES" (without quotes) in the only line.

Otherwise print "NO" (without quotes) in the first line.

Examples

input
3 586
output
NO

input
2 09
output
NO

input
9 123456789
output
YES

input
3 911
output
YES

Note

You can find the picture clarifying the first sample case in the statement above.

Uma academia de balé irá organizar uma Oficina de Balé Intensivo (OBI) na Semana de Balé Contemporâneo (SBC). Nessa academia, existem N bailarinas que praticam regularmente. O dono da academia, por ser experiente, consegue medir o nível de habilidade de cada uma delas por um número inteiro; nessa medição, números maiores correspondem a dançarinas mais habilidosas, e os números obtidos são todos distintos. Além disso, ele possui uma lista das bailarinas em ordem cronológica de ingresso na academia: As bailarinas que aparecem primeiro na lista estão há mais tempo na academia, e as que estão no final ingressaram mais recentemente.

O dono da academia decidiu escolher duas das bailarinas para ajudá-lo na realização do evento: uma ajudará no trabalho braçal, enquanto a outra irá exemplificar os passos de balé. Por seu perfeccionismo, ele deseja que a bailarina que exemplificará os passos de dança seja, dentre as duas meninas do par, a mais habilidosa e a que frequenta a academia há mais tempo.

Ele sabe que a Oficina será um sucesso desde que os dois critérios mencionados acima sejam satisfeitos pela dupla de dançarinas escolhidas. Com isso, ele ficou curioso para saber quantas duplas de dançarinas podem ajudá-lo na Oficina. A quantidade de dançarinas, contudo, é relativamente grande e ele não possui nem tempo nem paciência para fazer tal cálculo. Como vocês são amigos, ele pediu a sua ajuda para contar quantas duplas são válidas. Você pode ajudá-lo?

Por exemplo, digamos que a academia possua 5 dançarinas com níveis de habilidade 1, 5, 2, 4 e 3, onde a primeira, que possui nível "1", está na academia há mais tempo e a última, que possui nível "3", está há menos. Temos, então, 4 possíveis duplas que poderemos usar nesta Oficina, que são (5, 2), (5, 4), (5, 3) e (4, 3). Note que a dupla (1, 3), por exemplo, não pode ser escolhida pelo dono da academia, pois a mais habilidosa dentre as duas é também a mais nova da dupla.

Entrada

A primeira linha contém um número N , que representa a quantidade de dançarinas que estão registradas na academia. A segunda linha da entrada contém N inteiros, onde o primeiro inteiro é o nível da dançarina que está há mais tempo na academia, o segundo inteiro é o nível da próxima dançarina mais antiga na academia (mas mais nova que a dançarina anterior), e assim sucessivamente.

Saída

A saída consistirá num único número X , que representa o total de duplas de dançarinas válidas para essa Oficina, dadas as regras descritas anteriormente.

Restrições

- $1 \leq N \leq 100\,000$.
- Todas as dançarinas possuirão níveis distintos, entre 1 e 100 000.
- O total de pares válidos, em todos os casos, será $\leq 1\,000\,000$.

Exemplos

Entrada

5

1 5 2 4 3

Saída

4

Entrada

9

9 8 7 6 5 4 3 1 2

Saída

35

The Innovative Consumer Products Company (ICPC) is planning to start a top-secret project. The project consists of s subprojects. There will be $b \geq s$ branches of ICPC involved in this project as ICPC wants to assign each branch to one of the subprojects. In other words, the branches will form disjoint groups, with each group in charge of a subproject.

At the end of each month, each branch will send a message to every other branch in its group (different message to each branch). ICPC has a particular protocol for its communications. Each branch i has a secret key k_i known only to the branch and the ICPC headquarters. Assume branch i wants to send a message to branch j . Branch i encrypts its message with its key k_i . A trusted courier picks up this message from this branch and delivers it to the ICPC headquarters. Headquarters decrypts the message with key k_i and re-encrypts it with key k_j . The courier then delivers this newly encrypted message to branch j , which decrypts it with its own key k_j . For security reasons, a courier can carry only one message at a time.

Given a road network and the locations of branches and the headquarters in this network, your task is to determine the minimum total distance that the couriers will need to travel to deliver all the end-of-month messages, over all possible assignments of branches to subprojects.

Input

The input file contains several test cases, each of them as described below.

The first line of input contains four integers n , b , s , and r , where n ($2 \leq n \leq 5000$) is the number of intersections, b ($1 \leq b \leq n - 1$) is the number of branches, s ($1 \leq s \leq b$) is the number of subprojects, and r ($1 \leq r \leq 50000$) is the number of roads. The intersections are numbered from 1 through n . The branches are at intersections 1 through b , and the headquarters is at intersection $b + 1$. Each of the next r lines contains three integers u , v , and ℓ , indicating a one-way road from intersection u to a different intersection v ($1 \leq u, v \leq n$) of length ℓ ($0 \leq \ell \leq 10000$). No ordered pair (u, v) appears more than once, and from any intersection it is possible to reach every other intersection.

Output

For each test case, display the minimum total distance that the couriers will need to travel on a line by itself.

Sample Input

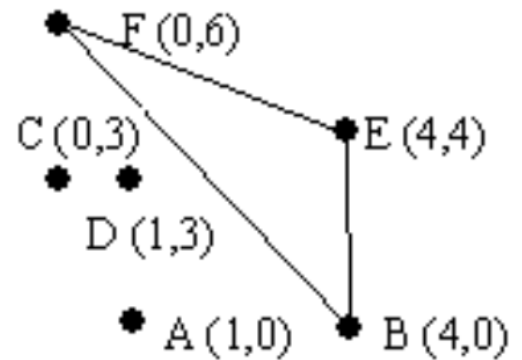
```
5 4 2 10
5 2 1
2 5 1
3 5 5
4 5 0
1 5 1
2 3 1
3 2 5
2 4 5
2 1 1
3 4 2
5 4 2 10
5 2 1
2 5 1
3 5 5
```

4 5 10
1 5 1
2 3 1
3 2 5
2 4 5
2 1 1
3 4 2

Sample Output

13
24

There has been considerable archeological work on the ancient Myacm culture. Many artifacts have been found in what have been called power fields: a fairly small area, less than 100 meters square where there are from four to fifteen tall monuments with crystals on top. Such an area is mapped out above. Most of the artifacts discovered have come from inside a triangular area between just three of the monuments, now called the power triangle. After considerable analysis archeologists agree how this triangle is selected from all the triangles with three monuments as vertices: it is the triangle with the largest possible area that does not contain any other monuments inside the triangle or on an edge of the triangle. Each field contains only one such triangle.



Archeological teams are continuing to find more power fields. They would like to automate the task of locating the power triangles in power fields. Write a program that takes the positions of the monuments in any number of power fields as input and determines the power triangle for each power field.

A useful formula: the area of a triangle with vertices (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) is the absolute value of

$$0.5 \times [(y_3 - y_1)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_1)].$$

Input

For each power field there are several lines of data. The first line is the number of monuments: at least 4, and at most 15. For each monument there is a data line that starts with a one character label for the monument and is followed by the coordinates of the monument, which are nonnegative integers less than 100. The first label is A, and the next is B, and so on.

There is at least one such power field described. The end of input is indicated by a 0 for the number of monuments. The first sample data below corresponds to the diagram in the problem.

Output

For each power field there is one line of output. It contains the three labels of the vertices of the power triangle, listed in increasing alphabetical order, with no spaces.

Sample Input

```
6
A 1 0
B 4 0
C 0 3
D 1 3
E 4 4
F 0 6
4
A 0 0
B 1 0
C 99 0
```

D 99 99
0

Sample Output

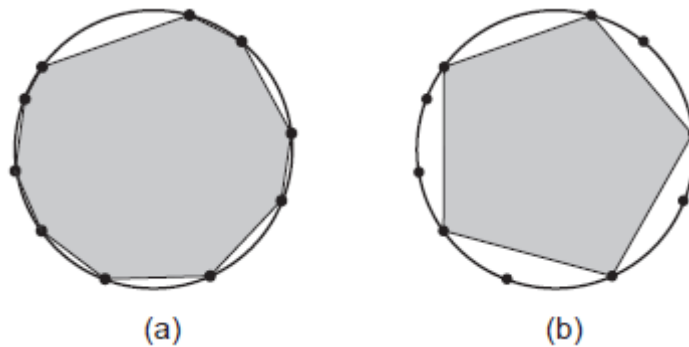
BEF
BCD

E. Encolhendo Polígonos

Time limit: 1s

Um polígono é dito inscrito em um círculo quando todos seus vértices estão naquele círculo. Nesse problema você receberá um polígono inscrito em um círculo, e você deve determinar o número mínimo de vértices que devem ser removidos para transformar o polígono dado em um polígono regular, i.e., um polígono que é equiangular (todos ângulos são congruentes) e equilátero (todos lados têm o mesmo comprimento).

Quando você remove um vértice v você primeiro remove o vértice e os segmentos de reta conectando-o aos seus vértices adjacentes w_1 e w_2 , e então você cria um novo segmento de reta conectando w_1 e w_2 . A figura (a) abaixo ilustra um polígono inscrito em um círculo, com dez vértices, e a figura (b) mostra um pentágono (polígono regular com cinco lados) formado ao remover cinco vértices do polígono em (a).



Nesse problema consideraremos que qualquer polígono deve ter pelo menos três lados.

Entrada

A entrada contém diversos casos de teste. A primeira linha de um caso de teste contém um inteiro N indicando o número de vértices do polígono inscrito ($3 \leq N \leq 10^4$). A segunda linha contém N inteiros X_i separados por espaços ($1 \leq X_i \leq 10^3$ para $0 \leq i \leq N - 1$). Cada X_i representa o comprimento do arco definido no círculo circunscrito, no sentido horário, pelos vértices i e $(i+1) \bmod N$. Lembre-se que um arco é um segmento da circunferência de um círculo; não o confunda com corda, que é um segmento de linha cujos ambos extremos estão no círculo.

O final da entrada é indicado por uma linha contendo apenas um zero.

Saída

Para cada caso de teste, seu programa deve imprimir uma única linha, contendo o número mínimo de vértices que precisam ser removidos do polígono dado para formar um polígono regular. Se não for possível formar um polígono regular, a linha deve conter apenas o valor -1.

Exemplo de Entrada

Exemplo de Saída

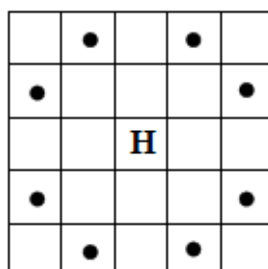
3	0
1000 1000 1000	2
6	-1
1 2 3 1 2 3	5
3	
1 1 2	
10	
10 40 20 30 30 10 10 50 24 26	
0	

ACM/ICPC South America Contest 2008.

F. Runner Pawns

Time limit: 1s

O jogo "Runner Pawns" é uma variante do clássico Xadrez que é jogado por apenas uma pessoa. O tabuleiro usado nesse jogo é semelhante ao tabuleiro do Xadrez, dividido em 8x8 quadrados. Como no Xadrez, cada quadrado pode conter apenas uma peça por vez. As peças do jogo são uma série de peões (os "Runner Pawns"), e um único cavalo, que é a única peça que o jogador comanda. O objetivo é capturar todos os peões antes de chegarem a ultima linha e se tornarem reis.



Possíveis movimentos do cavalo

Os movimentos do cavalo são ditos em forma de 'L', uma vez que o cavalo sempre anda dois quadrados em uma direção e mais um quadrado perpendicular a essa direção. A figura acima ilustra os movimentos do cavalo, onde o caractere 'H' indica a posição atual do cavalo e o caractere '.' indica um possível movimento do cavalo. Nota-se que a representação usando quadrados pretos e brancos do tabuleiro de Xadrez não é distinguida.

01 02 03 04 05 06 07 08
09 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

Da posição 22, o cavalo pode se mover para as posições 05, 07, 12, 16, 28, 32, 37 ou 39. Da posição 57, o cavalo pode se mover para as posições 42 ou 51.

O tabuleiro com as células numeradas

Os movimentos dos peões são um pouco diferentes do Xadrez, uma vez que o peão só pode se mover um quadrado para frente e todos os outros peões se movem ao mesmo tempo. Eles nunca se movem em diagonal. As casas do tabuleiro são numeradas de 1 a 64, como mostrado acima. Os peões se movem na direção vertical de cima para baixo, de modo que os quadrados numerados 57-64 são os objetivos dos peões.

Cada rodada do jogo é composta por um movimento do cavalo seguido por um movimento simultâneo de todos os peões ainda não capturados.

A fim de capturar um peão, o jogador deve mover o cavalo para o quadrado onde está o peão. O peão capturado deixa o tabuleiro e somente os peões restantes avançam para a próxima rodada. Para vencer o jogo, o jogador deve capturar todos os peões. Se um peão conseguir chegar a última linha, torna-se um rei. Com isso o cavalo tem apenas mais um movimento para captura-lo. Se isso não acontecer, o rei se move e isso significa que o jogo acaba e o jogador perde. Além disso, se o cavalo se mover para um quadrado que vai ser ocupado por um peão, no próximo movimento dos peões o cavalo é capturado pelo peão e o jogador perde.

Sua tarefa é escrever um programa que analise um tabuleiro de "Runner Pawns" e responda se existe uma sequencia de movimentos para o cavalo vencer. Se for possível, o programa deve determinar o número mínimo de movimentos necessários para o cavalo capturar todos os peões.

Entrada

A entrada contém várias instâncias do problema, uma por linha. Cada exemplo começa com um inteiro **P** que representa o número de peões ($0 \leq P \leq 8$), seguido por **P** inteiros ($1 \leq A_1, A_2, \dots, A_P \leq 64$) que descrevem a posição inicial de cada peão seguido de um inteiro **H** ($1 \leq H \leq 64$) que representa a posição inicial do cavalo. O fim da entrada é indicado quando **P** = 0.

Saída

Para cada instancia do problema da entrada, seu programa deve imprimir uma única linha, contendo a resposta para o problema. Se houver uma sequencia de movimentos para o cavalo que capture todos os peões (e sem que o cavalo seja capturado por um peão), o programa deve imprimir o comprimento da menor sequencia de movimentos possível. Caso contrário, seu programa deve imprimir a palavra 'impossivel'.

Exemplo de Entrada	Exemplo de Saída
1 1 11	1
1 60 1	impossible
2 33 60 54	3
0	

ACM/ICPC South America Contest 2003.

Por Sergio Gabriel Tavares  Brasil

G. Aggressive cows

Time limit: 2s

Memory limit: 1536 MB

Farmer John has built a new long barn, with N ($2 \leq N \leq 100,000$) stalls. The stalls are located along a straight line at positions x_1, \dots, x_N ($0 \leq x_i \leq 1,000,000,000$).

His C ($2 \leq C \leq N$) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, FJ want to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

Input

t – the number of test cases, then t test cases follows.

* Line 1: Two space-separated integers: N and C

* Lines 2.. $N+1$: Line $i+1$ contains an integer stall location, x_i

Output

For each test case output one integer: the largest minimum distance.

Example

Input:

```
1
5 3
1
2
8
4
9
```

Output:

```
3
```

Output details:

FJ can put his 3 cows in the stalls at positions 1, 4 and 8, resulting in a minimum distance of 3.

H. Restaurant Tables

Time limit: 1s

Memory limit: 256 MB

In a small restaurant there are a tables for one person and b tables for two persons.

It is known that n groups of people come today, each consisting of one or two people.

If a group consists of one person, it is seated at a vacant one-seater table. If there are none of them, it is seated at a vacant two-seater table. If there are none of them, it is seated at a two-seater table occupied by single person. If there are still none of them, the restaurant denies service to this group.

If a group consists of two people, it is seated at a vacant two-seater table. If there are none of them, the restaurant denies service to this group.

You are given a chronological order of groups coming. You are to determine the total number of people the restaurant denies service to.

Input

The first line contains three integers n , a and b ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq a, b \leq 2 \cdot 10^5$) — the number of groups coming to the restaurant, the number of one-seater and the number of two-seater tables.

The second line contains a sequence of integers t_1, t_2, \dots, t_n ($1 \leq t_i \leq 2$) — the description of clients in chronological order. If t_i is equal to one, then the i -th group consists of one person, otherwise the i -th group consists of two people.

Output

Print the total number of people the restaurant denies service to.

Examples

input
4 1 2 1 2 1 1
output
0

input
4 1 1 1 1 2 1
output
2

Note

In the first example the first group consists of one person, it is seated at a vacant one-seater table. The next group occupies a whole two-seater table. The third group consists of one person, it occupies one place at the remaining two-seater table. The fourth group consists of one person, he is seated at the remaining seat at the two-seater table. Thus, all clients are served.

In the second example the first group consists of one person, it is seated at the vacant one-seater table. The next group consists of one person, it occupies one place at the two-seater table. It's impossible to seat the next group of two people, so the restaurant denies service to them. The fourth group consists of one person, he is seated at the remaining seat at the two-seater table. Thus, the restaurant denies service to 2 clients.

I. Trapézio

Time limit: 0.3s

Memory limit: 1536 MB

A família Stropovich é conhecida mundialmente no mundo do circo por seus incríveis números de trapézio. É uma família numerosa, e todos os seus membros de várias gerações participam dos números de trapézio. Bem, quase todos os membros da família. Percival Stropovich, um garoto brilhante mas desajeitado, nunca pôde participar desses espetáculos, pois sua presença é certeza de desastre.

Mas finalmente o Sr. Postrovich Stropovich, patriarca da família, encontrou uma atividade para o desajeitado garoto, quando soube que Percival tinha ganho uma medalha na Olimpíada de Informática. Como Percival é um excelente programador, foi escalado para uma tarefa muito importante: verificar se é possível realizar o número dos sonhos de todos na família Stropovich. O número desejado é colocar todos os membros da família pendurados em um único trapézio, formando uma linha de trapezistas, de modo que o primeiro trapezista segure o segundo trapezista, o segundo trapezista segure o terceiro trapezista, e assim por diante, até o último trapezista da família.

Na família há trapezistas mais fortes e mais fracos, mais pesados e mais leves. Percival conhece o peso e a força (capacidade de aguentar o peso dos trapezistas abaixo dele) de cada trapezista de sua família. Sua tarefa é determinar uma ordem em que cada trapezista segura no máximo um peso menor ou igual à sua capacidade, de forma que todos os trapezistas da família fiquem pendurados em um mesmo trapézio.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A primeira linha do conjunto de testes contém um número inteiro N que indica o número de trapezistas ($1 \leq N \leq 10^5$). Os trapezistas são identificados por números de 1 a N .

Cada uma das N linhas seguintes contém um par de inteiros P e F que representam respectivamente o peso do trapezista ($1 \leq P \leq 10^4$) e sua força ($0 \leq F \leq 10^9$).

Saída

Se é possível executar o número de acordo com as condições descritas acima, seu programa deve imprimir, na saída padrão, uma lista com N inteiros, um em cada linha, representando a ordem em que os trapezistas devem estar pendurados no trapézio. O primeiro número da lista deve corresponder ao trapezista que está no trapézio (não é seguro por ninguém), o último número da lista ao trapezista que não segura ninguém. Se houver mais de uma ordem possível de trapezistas, imprima a que tem menor ordem lexicográfica. Caso não seja possível executar o número, seu programa deve imprimir uma linha contendo a palavra 'IMPOSSIVEL' (sem acento).

(Nota: a ordem lexicográfica da lista a_1, a_2, \dots, a_N é menor do que a da lista b_1, b_2, \dots, b_N se para algum $1 \leq i \leq N$ temos $a_i < b_i$ e o prefixo a_1, a_2, \dots, a_{i-1} é igual ao prefixo b_1, b_2, \dots, b_{i-1})

Exemplo

Entrada:

2

100 900

1000 100

Saída:

2

1

Entrada:

3

100 100

100 100

100 100

Saída:

IMPOSSIVEL

Entrada:

3

100 200

100 200

100 200

Saída:

1

2

3

J. Buying A House

Time limit: 2s

Memory limit: 256 MB

Zane the wizard had never loved anyone before, until he fell in love with a girl, whose name remains unknown to us.



The girl lives in house m of a village. There are n houses in that village, lining in a straight line from left to right: house 1, house 2, ..., house n . The village is also well-structured: house i and house $i + 1$ ($1 \leq i < n$) are exactly 10 meters away. In this village, some houses are occupied, and some are not. Indeed, unoccupied houses can be purchased.

You will be given n integers a_1, a_2, \dots, a_n that denote the availability and the prices of the houses. If house i is occupied, and therefore cannot be bought, then a_i equals 0. Otherwise, house i can be bought, and a_i represents the money required to buy it, in dollars.

As Zane has only k dollars to spare, it becomes a challenge for him to choose the house to purchase, so that he could live as near as possible to his crush. Help Zane determine the minimum distance from his crush's house to some house he can afford, to help him succeed in his love.

Input

The first line contains three integers n, m , and k ($2 \leq n \leq 100$, $1 \leq m \leq n$, $1 \leq k \leq 100$) — the number of houses in the village, the house where the girl lives, and the amount of money Zane has (in dollars), respectively.

The second line contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 100$) — denoting the availability and the prices of the houses.

It is guaranteed that $a_m = 0$ and that it is possible to purchase some house with no more than k dollars.

Output

Print one integer — the minimum distance, in meters, from the house where the girl Zane likes lives to the house Zane can buy.

Examples

input

```
5 1 20
0 27 32 21 19
```

output

```
40
```

input

```
7 3 50
62 0 0 0 99 33 22
```

output
30

input
10 5 100 1 0 1 0 0 0 0 0 1 1
output
20

Note

In the first sample, with $k = 20$ dollars, Zane can buy only house 5. The distance from house $m = 1$ to house 5 is $10 + 10 + 10 + 10 = 40$ meters.

In the second sample, Zane can buy houses 6 and 7. It is better to buy house 6 than house 7, since house $m = 3$ and house 6 are only 30 meters away, while house $m = 3$ and house 7 are 40 meters away.

K. Homem, elefante e rato

Time limit: 0.1s

Memory limit: 1536 MB

Um jogo muito popular na Nlogônia é o Homem, Elefante e Rato. Ele é tipicamente jogado com apenas dois jogadores, e funciona da seguinte forma: cada jogador secretamente escolhe um dos três símbolos e, após uma contagem regressiva, ambos revelam simultaneamente o símbolo escolhido através de sinais manuais, estendendo a sua frente uma das mãos sinalizando sua escolha.

O Homem é representado pela mão fechada, como a cabeça de um homem. O Elefante é representado pela mão aberta, exibindo os cinco dedos, como a pata do elefante nlogonense. Por fim, o Rato é representado pela mão fechada, com o dedo indicador e o dedo médio esticados, como as orelhas do pequeno animal.



Figura 1: Os três símbolos do jogo Homem, Elefante e Rato.

Para determinar o vencedor é muito simples: o Homem sempre perde para o Elefante (pois é esmagado debaixo de sua pata), o Elefante sempre perde para o Rato (pois tem medo dele e foge correndo) e o Rato sempre perde para o Homem (que espalha ratoeiras para capturá-lo). Se dois jogadores utilizarem o mesmo símbolo, ocorre um empate e joga-se novamente.

Os habitantes da Nlogônia, que são estrategistas natos de Homem, Elefante e Rato, utilizam a seguinte técnica no campeonato nacional, realizado todos os anos: começam sempre jogando Homem até o momento em que este símbolo causa empates com a maioria dos oponentes. Eles então trocam sua estratégia para o símbolo que ganha daquele que usavam anteriormente. Assim, os jogadores vão mudar de Homem para Elefante, depois para Rato, depois de volta a Homem.

Para auxiliar um famoso competidor estrangeiro de um jogo parecido, que é quase, mas não completamente diferente de Homem, Elefante e Rato, você irá desenvolver um programa que contabiliza quantos jogadores irão utilizar cada símbolo.

Suponha que todos os N jogadores são dispostos em fila e identificados pela sua posição, de 1 a N . Seu programa deverá processar M comandos, de dois tipos: mudança de símbolo e contar a frequência dos símbolos. Ambos os comandos recebem um intervalo contíguo de jogadores na fila a serem considerados.

Entrada

A entrada é composta por diversos casos de teste. Cada caso de teste começa com uma linha contendo dois inteiros N e M que representam, respectivamente, o número de jogadores no campeonato e o número de operações.

As próximas M linhas contêm cada uma a descrição de uma operação. Operações de mudança de estratégia serão representadas por uma linha da forma “M A B” onde A e B são inteiros. Os jogadores cuja estratégias serão alteradas são aqueles cuja posição na fila está entre A e B , inclusive.

Operações de contagem serão representadas por uma linha da forma “C A B” onde A e B são inteiros representando o intervalo de jogadores que deverão ser considerados na contagem. Levaremos em conta os jogadores cuja posição na fila está entre A e B , inclusive.

Saída

Para cada operação de contagem, imprima uma linha contendo três inteiros indicando respectivamente o número de símbolos Homem, Elefante e Rato que são usados pelos jogadores no intervalo dado. Imprima também uma linha em branco após cada caso de teste, inclusive após o ultimo caso de teste da entrada.

Restrições

- $1 \leq N \leq 10^5$
- $0 \leq M \leq 10^6$
- $1 \leq A \leq B \leq N$

Exemplos

Entrada

```
10 7
C 1 10
M 5 6
C 5 6
M 6 7
C 4 8
M 1 10
C 1 10
5 6
M 1 5
M 2 4
M 1 2
M 4 5
C 1 5
C 3 4
```

Saída

```
10 0 0
0 2 0
2 2 1
1 7 2

2 0 3
1 0 1
```

L. Find The Bone

Time limit: 2s

Memory limit: 256 MB

Zane the wizard is going to perform a magic show shuffling the cups.

There are n cups, numbered from 1 to n , placed along the x -axis on a table that has m holes on it. More precisely, cup i is on the table at the position $x = i$.

The problematic bone is initially at the position $x = 1$. Zane will confuse the audience by swapping the cups k times, the i -th time of which involves the cups at the positions $x = u_i$ and $x = v_i$. If the bone happens to be at the position where there is a hole at any time, it will fall into the hole onto the ground and will not be affected by future swapping operations.

Do not forget that Zane is a wizard. When he swaps the cups, he does not move them ordinarily. Instead, he teleports the cups (along with the bone, if it is inside) to the intended positions. Therefore, for example, when he swaps the cup at $x = 4$ and the one at $x = 6$, they will not be at the position $x = 5$ at any moment during the operation.



Zane's puppy, Inzane, is in trouble. Zane is away on his vacation, and Inzane cannot find his beloved bone, as it would be too exhausting to try opening all the cups. Inzane knows that the Codeforces community has successfully helped Zane, so he wants to see if it could help him solve his problem too. Help Inzane determine the final position of the bone.

Input

The first line contains three integers n , m , and k ($2 \leq n \leq 10^6$, $1 \leq m \leq n$, $1 \leq k \leq 3 \cdot 10^5$) — the number of cups, the number of holes on the table, and the number of swapping operations, respectively.

The second line contains m **distinct** integers h_1, h_2, \dots, h_m ($1 \leq h_i \leq n$) — the positions along the x -axis where there is a hole on the table.

Each of the next k lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) — the positions of the cups to be swapped.

Output

Print one integer — the final position along the x -axis of the bone.

Examples

input
7 3 4 3 4 6 1 2 2 5 5 7 7 1
output

1
input
5 1 2 2 1 2 2 4
output
2

Note

In the first sample, after the operations, the bone becomes at $x=2$, $x=5$, $x=7$, and $x=1$, respectively.

In the second sample, after the first operation, the bone becomes at $x=2$, and falls into the hole onto the ground.