2020-2021 ACM-ICPC, Asia Nanjing Regional Contest (XXI Open Cup, Grand Prix of Nanjing)

Statement is not available on English language

B. Baby's First Suffix Array Problem

5 seconds, 256 megabytes

A suffix array for string s of length n is a permutation sa of integers from 1 to n such that $s[sa_1..n], s[sa_2..n], \dots, s[sa_n..n]$ is the list of non-empty suffixes of s sorted in lexicographical order. The rank table for suffixex of s is a permutation $rank$ of integers from 1 to n such that $rank_{sa_i} = i$.

Kotori has a string $s = s_1 s_2 \dots s_n$. She would like to ask m queries. And in the i-th query, a substring $x = s[l_i..r_i]$ of s is given, Kotori would like to know the rank of suffix $s[k_i..r_i]$ of x.

Note $s[l..r]$ means the substring of s which starts from the l-th position and ends at the r-th position, both inclusive.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n, m \le 5 \times 10^4$) – the length of the string and the number of queries.

The second line contains a string s of length n consisting only of lowercase English letters.

Each of the next m lines contains three integers l_i, r_i and k_i ($1 \le l_i \le r_i \le n, l_i \le k_i \le r_i$) denoting a query.

It is guaranteed that neither the sum of n or the sum of m of all test cases will exceed 5×10^4.

Output

For each query output one line containing one integer denoting the answer.

input

```
2
10 4
baaabbabba
2 8 3
1 1 1
2 3 2
2 5 4
20 3
cccbccbadaacbbbcccab
14 17 16
3 20 17
17 20 18
```

output

```
2
1
2
3
4
15
3
```

C. Certain Scientific Railgun

1 second, 256 megabytes

Misaka Mikoto is the third-ranked Level 5 esper in *Academy City* and has been nicknamed *Railgun* due to her signature move. One day, several evil robots invade Academy City and Misaka is planning to terminate all of them.

Consider Academy City as a 2-dimensional plane. There are n robots in total and the position of the i-th robot is (x_i, y_i). Misaka will start moving from $(0, 0)$ and her railgun ability will terminate all robots sharing the same x- or y-coordinate with her. More formally, if Misaka is now located at (x_m, y_m), all robots whose $x_i = x_m$ or $y_i = y_m$ will be terminated.

As Misaka hates decimals and Euclidean geometry, she will only move from one integer point to another integer point and can only move horizontally (parallel to the x-axis) or vertically (parallel to the y-axis). As moving among the city is quite tiresome, Misaka asks you to calculate the minimum distance she has to move to terminate all robots.

Recall that an integer point is a point whose x-coordinate and y-coordinate are both integers.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$) indicating the number of robots.

For the following n lines, the i-th line contains two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$) indicating the position of the i-th robot.

It is guaranteed that the sum of n of all test cases will not exceed 10^5.

Output

For each test case output one line containing one integer indicating the minimum distance Misaka needs to move to terminate all robots.

input

```
3
2
0 1
1 0
4
1 1
-3 -3
4 -4
-2 2
4
1 100
3 100
-100 1
3 -100
```

output

```
0
8
4
```

For the second sample test case, Misaka should first go to $(0, 1)$, then to $(0, 2)$, then to $(0, -3)$, then to $(0, -4)$.

For the third sample test case, Misaka should first go to $(1, 0)$, then to $(1, 1)$, then to $(3, 1)$.

D. Degree of Spanning Tree

2 seconds, 256 megabytes

Given an undirected connected graph with n vertices and m edges, your task is to find a spanning tree of the graph such that for every vertex in the spanning tree its degree is not larger than $\frac{n}{2}$.

Recall that the degree of a vertex is the number of edges it is connected to.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($2 \le n \le 10^5$, $n - 1 \le m \le 2 \times 10^5$) indicating the number of vertices and edges in the graph.

For the following m lines, the i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) indicating that there is an edge connecting vertex u_i and v_i. Please note that there might be self loops or multiple edges.

It's guaranteed that the given graph is connected. It's also guaranteed that the sum of n of all test cases will not exceed 5×10^5, also the sum of m of all test cases will not exceed 10^6.

Output

For each test case, if such spanning tree exists first output "Yes" (without quotes) in one line, then for the following $(n-1)$ lines print two integers p_i and q_i on the i-th line separated by one space, indicating that there is an edge connecting vertex p_i and q_i in the spanning tree. If no valid spanning tree exists just output "No" (without quotes) in one line.

input

```
2
6 9
1 2
1 3
1 4
2 3
2 4
3 4
4 5
4 6
4 6
3 4
1 3
2 3
3 3
1 2
```

output

```
Yes
1 2
1 3
1 4
4 5
4 6
No
```

For the first sample test case, the maximum degree among all vertices in the spanning tree is 3 (both vertex 1 and vertex 4 has a degree of 3). As $3 \le \frac{6}{2}$ this is a valid answer.

For the second sample test case, it's obvious that any spanning tree will have a vertex with degree of 2, as $2 > \frac{3}{2}$ no valid answer exists.

E. Evil Coordinate

1 second, 256 megabytes

A robot is standing on an infinite 2-dimensional plane. Programmed with a string $s_1 s_2 \cdots s_n$ of length n, where $s_i \in \{\text{'U'}, \text{'D'}, \text{'L'}, \text{'R'}\}$, the robot will start moving from $(0, 0)$ and will follow the instructions represented by the characters in the string.

More formally, let (x, y) be the current coordinate of the robot. Starting from $(0, 0)$, the robot repeats the following procedure n times. During the i-th time:

- If $s_i = \text{'U'}$ the robot moves from (x, y) to $(x, y+1)$;
- If $s_i = \text{'D'}$ the robot moves from (x, y) to $(x, y-1)$;
- If $s_i = \text{'L'}$ the robot moves from (x, y) to $(x-1, y)$;
- If $s_i = \text{'R'}$ the robot moves from (x, y) to $(x+1, y)$.

However, there is a mine buried under the coordinate (m_x, m_y). If the robot steps onto (m_x, m_y) during its movement, it will be blown up into pieces. Poor robot!

Your task is to rearrange the characters in the string in any order, so that the robot will not step onto (m_x, m_y).

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers m_x and m_y ($-10^9 \le m_x, m_y \le 10^9$) indicating the coordinate of the mine.

The second line contains a string $s_1 s_2 \cdots s_n$ of length n ($1 \le n \le 10^5$, $s_i \in \{\text{'U'}, \text{'D'}, \text{'L'}, \text{'R'}\}$) indicating the string programmed into the robot.

It's guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

For each test case output one line. If a valid answer exists print the rearranged string, otherwise print "Impossible" (without quotes) instead. If there are multiple valid answers you can print any of them.

input

```
5
1 1
RURULLD
0 5
UUU
0 3
UUU
0 2
UUU
0 0
UUU
```

output

```
LDLRUUR
UUU
Impossible
Impossible
Impossible
```

Statement is not available on English language

Statement is not available on English language

H. Harmonious Rectangle

1 second, 256 megabytes

A vertex-colored rectangle is a rectangle whose four vertices are all painted with colors. For a vertex-colored rectangle, it's harmonious if and only if we can find two adjacent vertices with the same color, while the other two vertices also have the same color with each other.

For example, $\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ are harmonious, while $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is not (same number for same color, and different numbers for different colors).

For each point in $\{(x, y) | 1 \le x \le n, 1 \le y \le m, x, y \in \mathbb{Z}\}$, where $\mathbb{Z}$ is the set of all integers, Kotori wants to paint it into one of the three colors: red, blue, yellow. She wonders the number of different ways to color them so that there exists at least one harmonious rectangle formed by the points, whose edges are all parallel to the x- or y-axis. That is to say, there exists $1 \le x_1 < x_2 \le n$ and $1 \le y_1 < y_2 \le m$ such that

$$\begin{cases} \mathrm{color}(x_1, y_1) = \mathrm{color}(x_1, y_2) \\ \mathrm{color}(x_2, y_1) = \mathrm{color}(x_2, y_2) \end{cases}$$

or

$$\begin{cases} \mathrm{color}(x_1, y_1) = \mathrm{color}(x_2, y_1) \\ \mathrm{color}(x_1, y_2) = \mathrm{color}(x_2, y_2) \end{cases}$$

where $\mathrm{color}(x, y)$ is the color of point (x, y).

Two coloring plans are considered different if there exists a point having different colors in the two coloring plans.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first and only line contains three integers n, m($1 \le n, m \le 2 \times 10^3$).

Output

For each test case output one line containing one integer indicating the number of different ways of coloring modulo $(10^9 + 7)$.

input

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

output

```
0
15
16485
```

I. Interested in Skiing

1 second, 256 megabytes

Kotori is interested in skiing. The skiing field is an infinite strip going along y-axis on the 2-dimensional plane where all points (x, y) in the field satisfies $-m \le x \le m$. When skiing, Kotori cannot move out of the field, which means that the absolute value of his x-coordinate should always be no more than m. There are also n segments on the ground which are the obstacles and Kotori cannot move across the obstacles either.

Kotori will start skiing from $(0, -10^{10^{10^{10^{10}}}})$ (you can regard this y-coordinate as a negative infinity) and moves towards the positive direction of the y-axis. Her vertical (parallel to the y-axis) speed is always v_y which cannot be changed, however she can control her horizontal (parallel to the x-axis) speed in the interval of $[-v_x, v_x]$. The time that Kotori changes her velocity can be neglected.

Your task is to help Kotori calculate the minimum value of v_x^* that once $v_x > v_x^*$ she can safely ski through the skiing field without running into the obstacles.

Input

There is only one test case in each test file.

The first line of the input contains three positive integers n, m and v_y ($1 \le n \le 100$, $1 \le m \le 10^4$, $1 \le v_y \le 10$), indicating the number of obstacles, the half width of the skiing field and the vertical speed.

For the following n lines, the i-th line contains four integers x_1, y_1, x_2 and y_2 ($-m \le x_1, x_2 \le m$, $-10^4 \le y_1, y_2 \le 10^4$, $x_1 \ne x_2$ or $y_1 \ne y_2$) indicating the i-th obstacle which is a segment connecting point (x_1, y_1) and (x_2, y_2), both inclusive (that is to say, these two points are also parts of the obstacle and cannot be touched). It's guaranteed that no two obstacles intersect with each other.

Output

Output one line containing one number indicating the minimum value of v_x^*. If it is impossible for Kotori to pass through the skiing field, output "-1" (without quotes) instead.

Your answer will be considered correct if and only if its absolute or relative error does not exceed 10^{-6}.

input

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

output

```
1.000000000000000
```

input

```
2 1 2
-1 0 1 0
1 1 0 1
```

output

```
-1
```

input

```
2 3 7
-3 0 2 2
3 1 -2 17
```

output

```
1.866666666666666
```

input

```
1 100 1
-100 0 99 0
```

output

```
0.000000000000000
```

J. Just Another Game of Stones

3 seconds, 256 megabytes

Kotori and Umi are playing games of stones, which is hosted by Honoka. The rule is the same as the classic one: There are some piles of stones and the players take turns to remove any positive number of stones from one pile. The one who can't make a legal move loses the game.

This time however, things will be a little different. As the host, Honoka will prepare the games from n candidate piles of stones, where the i-th pile initially has a_i stones. Honoka will perform q operations of the following two types:

1. Given three integers l, r and x, for all $l \le i \le r$ change the number of stones in the i-th candidate pile to $\max(b_i, x)$, where b_i is the current number of stones in the i-th candidate pile.
2. Given three integers l, r and x, start a game of stones consisting of $(r-l+2)$ piles where the i-th pile contains b_{l-1+i} stones for all $1 \le i < (r-l+2)$, and the $(r-l+2)$-th pile contains x stones. Note that this operation is only querying for answer and will not affect the state of the n candidate piles of stones.

Kotori is always the first to move. As a big fan of Kotori, you would like to know, for each game of stones, the number of ways Kotori can play in the first step to ensure her victory if both players use the best strategy. We consider two ways different if Kotori is taking stones from different piles, or from the same pile but is taking different number of stones.

Input

There is only one test case in each test file.

The first line of the input contains two integers n and q ($1 \le n, q \le 2 \times 10^5$) indicating the number of candidate piles and the number of operations.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(0 \le a_i \le 2^{30} - 1)$ where a_i indicates the initial number of stones in the i-th pile.

For the following q lines, the i-th line contains four integers op_i, l_i, r_i and x_i ($op_i \in \{1, 2\}$, $1 \le l_i \le r_i \le n$, $0 \le x_i \le 2^{30} - 1$) indicating the i-th operation, where op_i is the type of operation and the others are the parameters of the operation. Operations are given in the order they're performed.

Output

For each operation of the second type output one line containing one integer indicating the answer.

input

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

output

```
1
0
3
```

For the first operation the players will play a game of stones consisting of 1, 2, 1 and 1 stone(s) in each pile respectively. The only winning play for Kotori is reduce the pile with 2 stones to 1 stone.

After the second operation, number of stones in the candidate piles changes to 1, 3, 3, 4 and 1 respectively.

For the fourth operation the players will play a game of stones consisting of 1, 3, 3, 4 and 4 stone(s) in each pile respectively. The winning plays for Kotori is to reduce the pile with 1 stone to 0 stone, or to reduce any pile with 3 stones to 2 stones.

K. K Co-prime Permutation

1 second, 256 megabytes

Kotori is very good at math (really?) and she loves playing with permutations and primes.

One day, she thinks of a special kind of permutation named k *co-prime permutation*. A permutation $p_1, p_2, \cdots, p_n$ of n is called a k co-prime permutation of n if there exists exactly k integers i such that $1 \le i \le n$ and $\gcd(p_i, i) = 1$, where $\gcd(x, y)$ indicates the greatest common divisor of x and y.

Given n and k, please help Kotori construct a k co-prime permutation of n or just report that there is no such permutation.

Recall that a permutation of n is a sequence of length n containing all integers from 1 to n.

Input

There is only one test case in each test file.

The first and only line contains two integers n and k ($1 \le n \le 10^6$, $0 \le k \le n$).

Output

Output one line containing n integers $p_1, p_2, \cdots, p_n$ separated by one space, indicating the permutation satisfying the given constraints. If no such permutation exists output "-1" (without quotes) instead. If there are multiple valid answers you can print any of them.

Please, DO NOT output extra spaces at the end of each line, otherwise your answer may be considered incorrect!

input

```
5 3
```

output

```
1 4 5 2 3
```

input

```
1 0
```

output

```
-1
```

L. Let's Play Curling

1 second, 256 megabytes

Curling is a sport in which players slide stones on a sheet of ice toward a target area. The team with the nearest stone to the center of the target area wins the game.

Two teams, Red and Blue, are competing on the number axis. After the game there are $(n+m)$ stones remaining on the axis, n of them for the Red team and the other m of them for the Blue. The i-th stone of the Red team is positioned at a_i and the i-th stone of the Blue team is positioned at b_i.

Let c be the position of the center of the target area. From the description above we know that if there exists some i such that $1 \le i \le n$ and for all $1 \le j \le m$ we have $|c - a_i| < |c - b_j|$ then Red wins the game. What's more, Red is declared to win p points if the number of i satisfying the constraint is exactly p.

Given the positions of the stones for team Red and Blue, your task is to determine the position c of the center of the target area so that Red wins the game and scores as much as possible. Note that c can be any real number, not necessarily an integer.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m $(1 \le n, m \le 10^5)$ indicating the number of stones for Red and the number of stones for Blue.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(1 \le a_i \le 10^9)$ indicating the positions of the stones for Red.

The third line contains m integers $b_1, b_2, \cdots, b_m$ $(1 \le b_i \le 10^9)$ indicating the positions of the stones for Blue.

It's guaranteed that neither the sum of n nor the sum of m will exceed 5×10^5.

Output

For each test case output one line. If there exists some c so that Red wins and scores as much as possible, output one integer indicating the maximum possible **score** of Red (NOT c). Otherwise output "Impossible" (without quotes) instead.

input

```
3
2 2
2 3
1 4
6 5
2 5 3 7 1 7
3 4 3 1 10
1 1
7
7
```

output

```
2
3
Impossible
```

For the first sample test case we can assign $c = 2.5$ so that the stones at position 2 and 3 for Red will score.

For the second sample test case we can assign $c = 7$ so that the stones at position 5 and 7 for Red will score.

M. Monster Hunter

1 second, 256 megabytes

There is a rooted tree with n vertices and the root vertex is 1. In each vertex, there is a monster. The hit points of the monster in the i-th vertex is hp_i.

Kotori would like to kill all the monsters. The monster in the i-th vertex could be killed if the monster in the direct parent of the i-th vertex has been killed. The power needed to kill the i-th monster is the sum of hp_i and the hit points of all other living monsters who lives in a vertex j whose direct parent is i. Formally, the power equals to

$$hp_i + \sum_{\substack{\text{the monster in vertex } j \text{ is \bf{alive}} \\ \text{and } i \text{ is the direct parent of } j}} hp_j$$

In addition, Kotori can use some magic spells. If she uses one magic spell, she can kill any monster using 0 power without any restriction. That is, she can choose a monster even if the monster in the direct parent is alive.

For each $m = 0, 1, 2, \cdots, n$, Kotori would like to know, respectively, the minimum total power needed to kill all the monsters if she can use m magic spells.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($2 \le n \le 2 \times 10^3$), indicating the number of vertices.

The second line contains $(n-1)$ integers $p_2, p_3, \cdots, p_n$ $(1 \le p_i < i)$, where p_i means the direct parent of vertex i.

The third line contains n integers $hp_1, hp_2, \cdots, hp_n$ $(1 \le hp_i \le 10^9)$ indicating the hit points of each monster.

It's guaranteed that the sum of n of all test cases will not exceed 2×10^3.

Output

For each test case output one line containing $(n+1)$ integers $a_0, a_1, \cdots, a_n$ separated by a space, where a_m indicates the minimum total power needed to kill all the monsters if Kotori can use m magic spells.

Please, DO NOT output extra spaces at the end of each line, otherwise your answer may be considered incorrect!

input

```
3
5
1 2 3 4
1 2 3 4 5
9
1 2 3 4 3 4 6 6
8 4 9 4 4 5 2 4 1
12
1 2 2 4 5 3 4 3 8 10 11
9 1 3 5 10 10 7 3 7 9 4 9
```

output

```
29 16 9 4 1 0
74 47 35 25 15 11 7 3 1 0
145 115 93 73 55 42 32 22 14 8 4 1 0
```