Problem A
Animal Farm

Time limit: 2 seconds

Memory limit: 2048 megabytes

Problem Description

On Animal Farm, the animals have rebelled against their human owner and have taken over the management of the farm. To ensure equality and fairness among all the animals, they have decided to create a new set of rules. However, as the new leaders, the pigs have started making changes to the rules to favor themselves.

The farm maintains a hierarchy of animals based on their species, with each animal assigned a specific influence level. This influence level, represented as a positive integer, determines the animal's priority in decision-making. Within a group, an animal can make decisions if it has the highest influence level among the members.

The pigs have a plan to maximize their collective influence in the leadership council by selecting a specific group of animals. Given a list of animals with their species and influence levels, you are tasked to form the most influential leadership council while adhering to the following rules:

1. Only one pig species is allowed in the council to avoid power conflicts among the pigs.
2. Every council member of non-pig species should have an influence level less than the influence level of the only pig's in the council.

Determine the maximum total influence levels of the council that can be formed under these rules.

Input Format

The first line contains an integer n, representing the number of animals. The next n lines each contain a string `species` and a positive integer `influence`:

- `species` is a string representing the species of the animal, e.g., "`pig`", "`horse`", "`cow`", etc.
- `influence` is an integer representing the influence level of the animal.

Output Format

Output a single integer, the maximum total influence levels of the leadership council that can be formed following the rules.

Technical Specification

- $1 \le n \le 10^5$.

- The length of `species` is at most 10.
- `species` consists of only English characters in lowercase.
- At least one animal's `species` is `pig`.
- `influence` is at most 10^8.

Sample Input 1

```
5
pig 10
horse 15
pig 5
cow 20
sheep 25
```

Sample Output 1

```
10
```

Sample Input 2

```
5
pig 10
horse 15
pig 15
cow 15
sheep 10
```

Sample Output 2

```
25
```

Problem B
Business Magic

Time limit: 2 seconds
Memory limit: 2048 megabytes

Problem Description

There are n stores located along a street, numbered from 1 to n from nearest to farthest. Last month, the store k had a net profit of r_k. If r_k is positive, it represents a profit of r_k dollars; if r_k is negative, it represents a loss of $-r_k$ dollars.

As a master of business magic, you have two types of spells at your disposal that you can use to alter the net profits of these stores for the next month:

1. **Blue Magic**: You can choose a single continuous interval $[L, R]$. The effect of this spell will be to double the net profit of every store from store L to store R (inclusive) for the next month. That is, if $k \in [L, R]$, then store k will have net profit $2r_k$ next month.
2. **Green Magic**: You can choose any store and cast the green magic on it. The effect of the green magic is to change the next month's net profit of that store to the negative of its last month's net profit.

Any store that has not been affected by either spell will have the same net profit next month as it did last month.

However, there are some restrictions when casting spells. You can only cast the blue magic once and it must be used before the green magic. Additionally, the green magic cannot be cast on any store that has already been affected by the blue magic. Your task is to determine the *maximum possible sum of the net profits* for all stores for the next month after casting your spells optimally.

Input Format

The first line contains an integer n, the number of stores. The second line contains n space-separated integers $r_1, r_2, \ldots, r_n$, where r_k is the net profit of store k last month.

Output Format

Output a single integer, the maximum possible total net profit of all stores for the next month after casting the spells optimally.

Technical Specification

- $1 \le n \le 3 \times 10^5$
- $-10^9 \le r_k \le 10^9$ for $k \in \{1, 2, \ldots, n\}$

Sample Input 1

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

Sample Output 1

```
20
```

Sample Input 2

```
7
-1 -1 -1 -1 -1 -1 -1
```

Sample Output 2

```
7
```

Sample Input 3

```
4
998244353 864197532 -7 1000000000
```

Sample Output 3

```
5724883756
```

Problem C
Cards

Time limit: 2 seconds
Memory limit: 2048 megabytes

Problem Description

Diana is a student who likes to play various types of board games. Today, she receives a deck of cards from her teacher as her birthday gift!

The deck of cards is special: there are n cards in the deck, and each card has a number on its front and another number on its back. Each number on the front or the back is an integer from 1 to n. Furthermore, all n numbers on the front are unique, and so are the n numbers on the back. In other words, numbers on the front and the back are two different permutations of numbers from 1 to n.

Apart from board games, Diana is also interested in mathematics and computer science. While she is playing with those cards, the concept of inversions in a permutation comes to her mind. An inversion is defined as a pair of indices (i, j) such that $i < j$ and the element at position i is greater than the element at position j. In other words, an inversion represents a situation where two elements are "out of order" relative to their positions. A permutation has inversion count c if there are c inversions can be found within it.

Diana wonders if she could rearrange the cards in some order so that the permutation on the front has the same inversion count as the permutation on the back (she cannot flip or throw away some cards). She cannot solve the problem in a while, so she wants to hear your solution.

In formal, you are given two permutations of integers from 1 to n: $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$. You have to find another permutation of the first n positive integers $p_1, p_2, \ldots, p_n$, such that $a' = [a_{p_1}, a_{p_2}, \ldots, a_{p_n}]$ and $b' = [b_{p_1}, b_{p_2}, \ldots, b_{p_n}]$ have the same inversion count. Output the sequences a' and b'.

Input Format

The first line of the input contains an integer n, denoting the number cards in the deck. The second line of the input contains n integers $a_1, a_2, \ldots, a_n$, where a_i is the number on the front of the i-th card. The third line of the input contains n integers $b_1, b_2, \ldots, b_n$, where b_i is the number on the back of the i-th card.

Output Format

If it is impossible to rearrange the cards so that the aforementioned condition is satisfied, print `No`. Otherwise, print `Yes` in the first line of the output. Then in the second line of the output, print n integers $a'_1, a'_2, \ldots, a'_n$, denoting the numbers on the front of the cards after the rearrangement. In the third line of the output, print n integers $b'_1, b'_2, \cdots, b'_n$, denoting the numbers on the back of the cards after the rearrangement.

If there are multiple possible solutions, print any of them.

Technical Specification

- $1 \le n \le 5 \times 10^5$
- $1 \le a_i \le n$ for $i \in \{1, 2, \ldots, n\}$
- $1 \le b_i \le n$ for $i \in \{1, 2, \ldots, n\}$
- It is guaranteed that $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ are both permutations of integers $1, 2, \ldots, n$.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

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

Sample Output 2

```
No
```

Sample Input 3

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

Sample Output 3

```
Yes
2 3 8 1 4 5 9 6 7 10
1 2 3 9 6 7 4 5 8 10
```

Sample Input 4

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

Sample Output 4

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

Problem D
Disbursement on Quarantine Policy

Time limit: 2 seconds

Memory limit: 2048 megabytes

Problem Description

> The 2019 novel coronavirus, COVID-19, can be transmitted between humans through water droplets and close contact. The transmission is especially easy and fast in relatively crowded or confined spaces, such as airplanes or trains. If someone is infected with COVID-19, then passengers occupying the adjacent seats will be infected easily.
>
> —"Quarantine Policy," 2023 ICPC Taoyuan Regional Contest, Problem D

There is a train with n rows, and there are m seats per row. All seats are occupied. For some passengers, we know they are being infected with COVID-19 or not. However, for other passengers, we are not sure about their status, and we assume each of them has $\frac{1}{2}$ chance being infected with COVID-19, independent from each other.

All infected passengers need to be quarantined for d_0 days. All passengers that are not infected, but edge-adjacent to any infected passenger, need to be quarantined for d_1 days. All passengers that are not infected, not edge-adjacent to any infected passenger, but corner-adjacent to any infected passenger, need to be quarantined for d_2 days.

The passengers need to stay in the hotel during quarantine. According to the regulations, the government needs to pay for the hotel. As an accountant of the government, you are asked to evaluate the expected total number of days the passengers need to be quarantined, which indicates the expected total cost on paying for the hotel.

For example, suppose $n = 4$, $m = 4$, $d_0 = 15$, $d_1 = 7$, $d_2 = 3$. The third passenger in the third row is infected, and we don't know whether the second passenger in the first row is infected or not. Other 14 passengers are not infected.

If the second passenger in the first row is infected, then the total number of days of quarantine is 91:

```
7 15  7  0
3  7  7  3
0  7 15  7
0  3  7  3
```

If that passenger is not infected, then the total number of days of quarantine is 55:

```
0  0  0  0
0  3  7  3
0  7 15  7
0  3  7  3
```

So the expected total number of days of quarantine is $\frac{91+55}{2} = 73$.

Input Format

The first line contains five integers n, m, d_0, d_1 and d_2. The following n lines contain m characters each. The j-th character of the i-th line represents the passenger occupying the j-th seat of the i-th row. Each character is one of '`V`', '`.`', or '`?`'. '`V`' means an infected passenger, '`.`' means a not infected passenger, and '`?`' means a passenger that has $\frac{1}{2}$ chance being infected.

Output Format

The expected total number of days the passengers need to be quarantined, modulo $10^9 + 7$. It can be proved that the answer can be represented by a rational number $\frac{p}{q}$ where q is not a multiple of $10^9 + 7$. Then you need to print $p \times q^{-1}$ modulo $10^9 + 7$, where q^{-1} means the multiplicative inverse of q modulo $10^9 + 7$.

Note: If $x \times q \equiv 1 \mod 10^9 + 7$, then x is the multiplicative inverse of q modulo $10^9 + 7$.

Technical Specification

- $1 \le n \le 100$
- $1 \le m \le 100$
- $0 \le d_2 \le d_1 \le d_0 \le 100$

Sample Input 1

```
4 4 15 7 3
.?..
....
..V.
....
```

Sample Output 1

```
73
```

Sample Input 2

```
2 2 1 1 1
??
??
```

Sample Output 2

```
750000009
```

Problem E
Efficient Slabstones Rearrangement

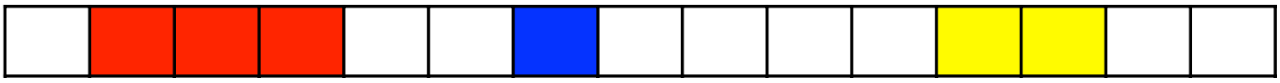
Time limit: 2 seconds

Memory limit: 2048 megabytes

Problem Description

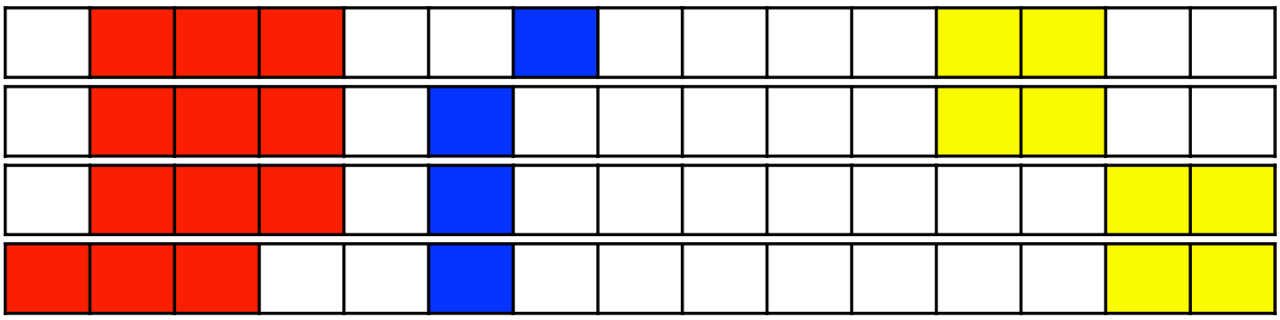
Barbara has a garden. The garden is long and narrow, divided into m equal-sized regions arranged in a row. Her friend, Babara, gave her n slabstones as birthday present. Barbara then placed these slabstones in her garden, so she can enjoy stepping slabstones from one to another every day. The i-th slabstone fully occupies the l_i-th to r_i-th region of the garden. The slabstones do not overlap, and any two slabstones have at least d empty regions between them.

Below is a valid placement of the slabstones with $m = 15$, $n = 3$, $d = 2$, and the three slabstones occupy the regions 2–4, 7–7, 12–13 respectively.



Barbara recently bought another slabstone that will occupy x consecutive regions in her garden. She will shift the original slabstones within the garden, then place the new slabstone somewhere in the garden. **After shifting** the original slabstones and placing the new slabstone, the slabstones cannot overlap, and any two slabstones must have at least d empty regions between them. The slabstones should remain non-overlapping **during slabstone rearrangement**.

Please note that, two slabstones can have less than d regions between them during slabstone rearrangement. For example, the following process is valid when $d = 2$:



Shifting a single slabstone to an adjacent region takes one minute. For example, the above rearrangement process takes 4 minutes. Now Barbara wants to know the minimum possible total time required to rearrange the slabstones, so she can save time for "other purposes".

Input Format

The first line contains four integers n, m, d and x. The i-th of the following n lines contains two integers l_i and r_i.

Output Format

The minimum possible total time (in minutes) to rearrange the slabstones so the new slabstone can be placed in the garden. If the new slabstone cannot be placed in the garden no matter how the slabstones are rearranged, just output `-1`.

Technical Specification

- $1 \le n \le 2000$
- $1 \le d \le m \le 10^9$
- $1 \le x \le m \le 10^9$
- $1 \le l_i \le r_i \le m$ for $i \in \{1, 2, \ldots, n\}$
- $r_i + d + 1 \le l_{i+1}$ for $i \in \{1, 2, \ldots, n-1\}$. That is, the slabstones are given in order from left to right.

Sample Input 1

```
3 15 2 3
2 4
7 7
12 13
```

Sample Output 1

```
4
```

Sample Input 2

```
5 100 1 75
2 3
5 7
11 13
17 19
23 29
```

Sample Output 2

```
9
```

Sample Input 3

```
1 100 99 1
1 1
```

Sample Output 3

```
-1
```

Problem F
Fibonacci Lucky Numbers

Time limit: 2 seconds

Memory limit: 2048 megabytes

Problem Description

Welcome to the **Lucky 777 Slot Game**! This game is known for its complex mathematical challenges, where only the smartest can win the jackpot.

The slot machine is powered by a mysterious sequence—the **Fibonacci sequence**. But it's no ordinary Fibonacci sequence; it has a twist inspired by the number 7, the symbol of luck in slot games.

When you pull the lever of the **Lucky 777 Slot Machine**, it generates a gigantic number using an integer n and the power of sevens: $7^{7^{7^n}}$. This number, however, is so massive that even the most powerful computers cannot handle it directly.

To claim the jackpot, you need to compute the last 10 digits of the $F_{7^{7^{7^n}}}$, the $7^{7^{7^n}}$-th Fibonacci number.

Input Format

The first line contains an integer t indicating the number of test cases. Each of the following t lines is a test case and contains exactly one positive integer n.

Output Format

For each test case, output one line contains the last 10 digits of $F_{7^{7^{7^n}}}$.

Technical Specification

- $1 \leq t \leq 20$
- $1 \leq n \leq 10^9$

Sample Input 1

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

Sample Output 1

```
1353646637
3172443437
2364206637
9010523437
9481646637
```

Note

The Fibonacci sequence is defined as:

- $F_0 = 0$
- $F_1 = 1$
- $F_k = F_{k-1} + F_{k-2}$ for $k \geq 2$

Problem G
Game of Rounding

Time limit: 2 seconds
Memory limit: 2048 megabytes

Problem Description

Jack got a new video game called "Rounding," which contains n levels. The game features a global ranking system that ranks all players worldwide based on their scores. Jack wants to break the global record and let everyone know who the master of this game is, so he has investigated the scoring system extensively.

He finally understands the scoring rules: when a player finishes each level, they earn some points. The player's score is the average points they earn per level, rounded to the nearest whole number. More precisely, if a player plays a total of k levels and earns $p_1, p_2, \ldots, p_k$ points respectively, their score will be $\lfloor \frac{\sum_{i=1}^{k} p_i}{k} + 0.5 \rfloor$. For example, if a player earns $[2, 3, 3]$ points in 3 levels, their score will be $\lfloor \frac{2+3+3}{3} + 0.5 \rfloor = 3$.

Jack has practiced several times and knows the points a_i he will earn in the i-th level. He discovered an exploit in the game that allows him to skip some levels at the beginning and stop at any time. This means Jack can choose a pair of numbers (l, r) where $1 \leq l \leq r \leq n$, and only play the levels from l to r.

Jack is curious about the maximum score he can achieve for each starting level l for $1 \leq l \leq n$, and how many levels he should play to achieve that maximum score. If there are several answers that yield the maximum score, he should print the smallest number of levels, as playing the game for a long time is unhealthy.

Input Format

The first line contains an integer t, indicating the number of test cases. Each test case consists of two lines. The first one contains an integer n, indicating the number of levels in the video game. The second one contains n space-separated integers, $a_1, a_2, \ldots, a_n$, representing the points Jack will earn in each level.

Output Format

For each test case, output n integers in one line. The i-th number indicates the number of levels Jack should play, starting from level i, to achieve the maximum score. If there are several answers that achieve the maximum score, print the smallest number of levels.

Technical Specification

- $1 \leq t \leq 10^5$

- $1 \leq n \leq 2 \times 10^5$
- $0 \leq a_i \leq 10^9$ for $i \in \{1, 2, \ldots, n\}$.
- The sum of n's of all test cases is at most 2×10^5.

Sample Input 1

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

Sample Output 1

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

Problem H
Harmonious Passage of Magicians

Time limit: 2 seconds
Memory limit: 2048 megabytes

Problem Description

There is a very narrow alley, and two teams of magicians want to pass through this alley from opposite ends. They do not see the other team until there is only a space that can hold one person between the two teams. Because the alley is so narrow, they cannot turn around or walk backward to avoid falling. However, being magicians, they can use a spell to teleport a short distance, allowing them to pass by another person. Additionally, to maintain order, the magicians in the same team cannot change their order, so they cannot use this spell to pass the magician which is from the same team.

To clarify, we assume that there are n magicians in the first team, starting from the left side and numbered from 1 to n, and m magicians in the second team, starting from the right side and numbered from $n+1$ to $n+m$.

The narrow alley has a total of $n+m+1$ spaces. The leftmost n spaces are occupied by the first team, facing right, and the rightmost m spaces are occupied by the second team, facing left. The alley configuration will look like this: $[1, 2, \ldots, n, \text{space}, n+1, n+2, \ldots, n+m]$.

When a magician moves, he must follow these rules:

- If there is an empty space directly in front of him, he can walk into that space.
- If there is a magician from the opposite team directly in front of him, and there is an empty space directly behind this magician, he can use the spell to move to that space.

Ultimately, the first team will occupy the rightmost n spaces, and the second team will occupy the leftmost m spaces.

To help them pass the alley, please provide a movement strategy that will allow them to pass. The strategy will be described with a sequence of numbers $a_1, a_2, \ldots$, where a_i indicates that in the i-th step, the magician with number a_i will move to an unoccupied space.

If there are multiple strategies, please output the lexicographically smallest one. Lexicographical order is a way of comparing strings or sequences of elements based on their alphabetical or numerical order. In the context of this problem, the "lexicographically smallest" strategy refers to the strategy that comes first in the numerical order when the strategies are represented as sequences of numbers.

More concretely:

- Each strategy is represented as a sequence of numbers: $a_1, a_2, \ldots$
- Two strategies are compared element by element:
  - If the first element of one strategy is smaller than the first element of the other, the first strategy is lexicographically smaller.
  - If the first elements are equal, compare the second elements, and so on.

Input Format

The first line contains an integer t, indicating the number of test cases. For the following t lines, each line contains two integers n and m, indicating the number of magicians from the first team and the number of magicians from the second team, respectively.

Output Format

Output t lines. The i-th line should contain the movement strategy that will help the magicians pass through the narrow alley for the i-th test case. If there are multiple strategies, output the lexicographically smallest one.

Technical Specification

- $2 \leq n \leq 3000$
- $2 \leq m \leq 3000$
- $1 \leq t \leq 1000$
- The sum of n's among all test cases is no more than 3000.
- The sum of m's among all test cases is no more than 3000.

Sample Input 1

```
2
2 2
2 3
```

Sample Output 1

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

Problem I
In Search of the Lost Array

Time limit: 2 seconds

Memory limit: 2048 megabytes

Problem Description

In a forgotten realm, a group of adventurers stumbles upon a set of mysterious scrolls hidden deep within an ancient library. These scrolls hold the secrets of a powerful numerical array that controls the magic of the realm. However, the scrolls have been damaged over time, and only fragments remain. Specifically, the adventurers discover a sequence of numbers representing the products of adjacent elements of an unknown array A.

The original array A consists of n integers $a_1, a_2, \ldots, a_n$ where $1 \le a_i \le 100$ for $1 \le i \le n$. The only information remaining on the scrolls is a sequence of $n-1$ integers $b_1, b_2, \ldots, b_{n-1}$, which are unordered products of adjacent elements from A. In other words:

$$\{b_1, b_2, \ldots, b_{n-1}\} = \{a_1 \times a_2, a_2 \times a_3, \ldots, a_{n-1} \times a_n\}$$

Your task is to help the adventurers reconstruct one possible original array A. If there are multiple valid arrays A that could result in the same sequence b, you may output any of them.

Input Format

The first line contains a single integer n, representing the length of the array A. The second line contains $n-1$ space-separated integers $b_1, b_2, \ldots, b_{n-1}$, representing the products of adjacent elements in the array A.

Output Format

If there is no such array A, then print `No` on a line. Otherwise, print `Yes` on the first line. Then, output n space-separated integers $a_1, a_2, \ldots, a_n$ on the second line, where $\{b_1, b_2, \ldots, b_{n-1}\} = \{a_1 \times a_2, a_2 \times a_3, \ldots, a_{n-1} \times a_n\}$.

Technical Specification

- $1 < n \le 18$.
- $1 \le a_i \le 100$ for $i \in \{1, 2, \ldots, n\}$
- $1 \le b_i \le 10000$ for $i \in \{1, 2, \ldots, n-1\}$

Sample Input 1

```
8
42 32 84 54 48 40 16
```

Sample Output 1

```
Yes
5 8 4 21 2 8 6 9
```

Sample Input 2

```
6
45 4 5 4 3
```

Sample Output 2

```
Yes
3 1 4 1 5 9
```

Sample Input 3

```
2
3246
```

Sample Output 3

```
No
```

Problem J
Just Round Down

Time limit: 2 seconds

Memory limit: 2048 megabytes

Problem Description

The Taiwan Online Programming Contest is a prestigious event that attracts talented programmers from all over the world. Known for its challenging problems and competitive environment, the contest has become a platform where only the best can prove their skills. However, there is one problem setter who has gained a notorious reputation among participants. This problem setter, known only by their pseudonym "`truckski`," has an unusual fascination with numbers – particularly big numbers, floating-point numbers, and any kind of mathematical challenge that involves precise calculations. `truckski` has a unique style of creating problems that often requires competitors to think carefully about the properties of numbers and how they can be manipulated.

In this year's edition of the Taiwan Online Programming Contest, `truckski` has come up with a seemingly simple yet tricky problem. The problem revolves around a fundamental concept in mathematics: rounding down a floating-point number to its nearest integer. While this task might appear straightforward at first glance, `truckski`'s twist lies in the precision required and the ability to handle a variety of floating-point values accurately.

Your task is to help the participants solve this problem by writing a program that takes a positive floating-point number as input and outputs the result of rounding it down to the nearest integer. This process is often referred to as taking the "floor" of a number. The floor of a number is the greatest integer that is less than or equal to the number itself.

Input Format

The input consists of a single line containing one positive floating-point number x.

Output Format

The output should be a single integer, which is the floor of the input number x. Please do not output decimal points.

Technical Specification

- $0 < x \leq 10^8$
- The input contains several digits and exactly one decimal point.
- The last printable character of the input must be a digit.
- There is at least one digit before the decimal point.
- There is no leading zero for $x \geq 1$.

- The size of input file is no more than 15 bytes.

Sample Input 1

```
1999.99
```

Sample Output 1

```
1999
```

Sample Input 2

```
2.00000
```

Sample Output 2

```
2
```

Note

The problem description is a fiction written by ChatGPT.

Problem K
Kingdom's Development Plan

Time limit: 2 seconds
Memory limit: 2048 megabytes

Problem Description

The Kingdom of Topcaria is planning a series of developmental projects to enhance its infrastructure. Each project has specific prerequisites that must be completed before the project can start. The Ministry of Development has asked you to help determine a feasible order in which all the projects can be completed.

You are given:

- n, the number of projects numbered from 1 to n.
- m, the number of prerequisite relationships between these projects.
- A list of m pairs, where each pair (a, b) indicates that project a must be completed before project b can start.

Your task is to determine an order in which all the projects can be completed. If it is impossible to complete all projects due to a cyclic dependency, output "`IMPOSSIBLE`". If there are multiple valid orders, please output any the lexicographically smallest one.

Input Format

The first line contains two integers n and m — the number of projects and the number of prerequisite relationships. The next m lines each contain two integers a and b — a prerequisite pair indicating that project a must be completed before project b.

Output Format

If it is not possible, output "`IMPOSSIBLE`". If it is possible to complete all projects, output a single line with n integers — a valid order of project completions. If there are multiple possible orders, output the lexicographically smallest one. An order is lexicographically smaller than another order if at the first position where they differ, the project number on the first order is smaller than the number on the second order.

Technical Specification

- $1 \le n \le 10^5$
- $0 \le m \le 2 \times 10^5$
- $a, b \in \{1, 2, \ldots, n\}$
- $a \ne b$

- No duplicate pairs are given.

Sample Input 1

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

Sample Output 1

```
1 2 3 4 5
```

Sample Input 2

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

Sample Output 2

```
IMPOSSIBLE
```

Problem L
Lexicopolis

Time limit: 2 seconds
Memory limit: 2048 megabytes

Problem Description

Welcome to Lexicopolis, the ancient city of legends and treasures. The city is famous for its intricate network of one-way roads. There are n intersections and m one-way roads connecting the intersections. People can only travel from intersection u_i to intersection v_i along road i, and road i is associated with a magical number w_i. A path of length k from intersection s to t is a sequence of roads $e_1, e_2, \ldots, e_k$ that allows travel from intersection s to intersection t. A path is lexicographically smaller than another path if at the first road where they have different magic numbers (not index), the number on the first path is smaller than the number on the second path.

It is rumored that the tourist who figures out the lexicographically smallest path of length k from intersection s to intersection t can receive a gift from the Lexicopolis government. Please write a program to find the lexicographicall smallest path of length k from intersection s to t. If it is impossible to travel from intersection s to t with exactly k roads, output `-1`.

Input Format

The first line contains six integers n, m, s, t, x, k. n is the number of intersections. m is the number of roads. s is the starting intersection and t is the ending intersection. x is a number that will be used for outputting the answer. k is the length of path. The i-th of the m following lines contains three integers u_i, v_i and w_i. That means road i is from intersection u_i to intersection v_i and associated with magic number w_i.

Output Format

If there is no path of length k from intersection s to t, output `-1`. Otherwise, assume such a path exists. Consider the lexicographically smallest path $e_1, e_2, \ldots, e_k$, and output $\sum_{i=1}^{k} w_{e_i} x^{k-i}$ modulo 10^9+7, where x is the number provided as the fifth value in the first line of the input.

Technical Specification

- $2 \le n \le 50$
- $1 \le m \le n^2 - n$
- $1 \le u_i \le n$ for $i \in \{1, 2, \ldots, m\}$
- $1 \le v_i \le n$ for $i \in \{1, 2, \ldots, m\}$
- $1 \le w_i \le 10^9$ for $i \in \{1, 2, \ldots, m\}$
- $u_i \ne v_i$ for $i \in \{1, 2, \ldots, m\}$

- $(u_i, v_i) \neq (u_j, v_j)$ for $i \neq j$
- $1 \leq s \leq n$
- $1 \leq t \leq n$
- $1 \leq k \leq 10^9$
- $1 \leq x \leq 10^9$

Sample Input 1

```
3 6 1 3 10 4
1 2 2
2 1 1
1 3 1
3 1 2
2 3 1
3 2 2
```

Sample Output 1

```
1211
```

Sample Input 2

```
3 6 1 3 10 5
1 2 2
2 1 1
1 3 1
3 1 2
2 3 1
3 2 2
```

Sample Output 2

```
12121
```

Sample Input 3

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

Sample Output 3

```
121513477
```

Sample Input 4

```
6 7 1 6 123 2
1 2 1000000000
2 4 2
```

Sample Output 4

```
-1
```

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