2025 ICPC Gran Premio de Mexico 1ra Fecha

A. Analysing Electrocardiograms

1 second, 256 megabytes

Dr. Cristina Yang is one of the best cardiologists in the world and you are her new student! To get you started, she has given you an easy task. You'll be given an example of how the heartbeat of a healthy heart looks in the form of a string S, and N electrocardiograms, each one is a string P, your task is to say if it's healthy or not. A healthy electrocardiogram is one which is made by concatenating multiple times the string S.

Input

In the first line a string S ($1 \leq |S| \leq 100$) the example of the healthy heart. In the second line a number N ($1 \leq N \leq 10^3$)the number of electrocadiograms you have to examine. In the next N lines a string P ($1 \leq |P| \leq 10^5$), the electrocardiogram to examine. It is assured that the sum of the lengths of the N strings is less than 10^5, and all strings are made of uppercase english characters.

Output

Print N lines, each one having either "`Yes`" or "`No`", corresponding to if the heart is healthy or not.

input
AB 3 ABAB ABA C
output
Yes No No

B. Buying Paint

1 second, 256 megabytes

Osvaldo and his friend Rogelio are looking to buy some paints to add some color to his house. He knows each color he uses to paint his house will add a beauty x (not necessarily positive) to the house. The store is having a special offer; the paints are placed in a square matrix, and if you buy a set of paints such that you can draw a spiral in the matrix beginning in one of the corners and filling the whole matrix, where the set is contiguous in the spiral, the paint is free! Since Rogelio is in a hurry to start training for ICPC, they have asked you to help them find the best sum of beauty one can add to the house if the paint is free, notice you can decide to buy no paints.

Input

On the first line a number N ($1 \leq N \leq 1000$), the length of the matrix. In the next N lines, N numbers $a_{i,j}$ ($-10^7 \leq a_{i,j} \leq 10^7$) indicating the beauty that color will add to the house if bought.

Output

Just a number X the best sum of beauty you can get added to the house such that the paints are free.

input
2 -1 3 2 -5
output
4

input
3 1 -2 4 3 8 -6 5 1 -10
output
20

A spiral is a traversal of the matrix that is a reflection or rotation of one similar to this:

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

This is the specific case for $n = 4$

C. Celayas New Sign

2 seconds, 256 megabytes

The city of Celaya is installing new street signs, and the mayor has appointed you to supervise the work. You have received a sign containing a string A made of english uppercase characters, and you suspect it might be incorrect. However, you are clever and have discovered that you can perform the following operation on the sign without damaging it:

1. First, partition the string A into four contiguous substrings P_1, P_2, P_3, P_4. For example, for the text `ABCD`, the partitions

   $$\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}$$

   and

   $$\texttt{AB}, \varepsilon, \texttt{CD}, \varepsilon$$

   are valid, while

   $$\texttt{A}, \texttt{C}, \texttt{BD}, \varepsilon$$

   is not. We denote the empty string as ε.
2. Then, create the string

   $$A' = P_1\, P_3\, P_2\, P_4$$

   by concatenation, and replace A with A'.

Unfortunately, there is not much time left before the deadline, so you may apply the operation **at most once** on the sign.

Given the current string on the sign and the correct one, determine whether it is possible to fix the sign or not.

Input

The first line of input contains the string A ($1 \leq |A| \leq 10^5$), the string that's written on the sign.

The second line of input contains the string B ($|A| = |B|$), the string that should be written on the sign.

It is guaranteed that the strings A and B have the same length and they consist of capital english letters.

Output

On the first line, print the word `Yes` if it is possible to fix the sign performing at most one operation, or the word `No` otherwise.

If it is possible to fix the sign, print three integers $l\ r\ k$ ($0 \leq l \leq r < |A|$, $0 \leq k \leq r - l$) on an additional line, which mean that we can fix the sign by taking:

$$\begin{aligned} P_1 &= A[0, l-1] \\ P_2 &= A[l, r-k] \\ P_3 &= A[r-k+1, r] \\ P_4 &= A[r+1, |A|-1] \end{aligned}$$

Where $A[x, y]$ with $x > y$ represents an empty string.

If there are many possible answers, print any of them.

input
ABC ACB
output
Yes 1 2 1

For the sample case, the string "ABC" can be transformed into the following strings:

- "ABC", with $l = 0, r = 0, k = 0$, since:
  - $P_1 = A[0, -1] = \varepsilon$
  - $P_2 = A[0, 0] = \texttt{A}$
  - $P_3 = A[1, 0] = \varepsilon$
  - $P_4 = A[1, 2] = \texttt{BC}$
- "CAB", with $l = 0, r = 2, k = 1$, since:
  - $P_1 = A[0, -1] = \varepsilon$
  - $P_2 = A[0, 1] = \texttt{AB}$
  - $P_3 = A[2, 2] = \texttt{C}$
  - $P_4 = A[3, 2] = \varepsilon$
- "BCA", with $l = 0, r = 2, k = 2$, since:
  - $P_1 = A[0, -1] = \varepsilon$
  - $P_2 = A[0, 0] = \texttt{A}$
  - $P_3 = A[1, 2] = \texttt{BC}$
  - $P_4 = A[3, 2] = \varepsilon$
- "ACB", with $l = 1, r = 2, k = 1$, since:
  - $P_1 = A[0, 0] = \texttt{A}$
  - $P_2 = A[1, 1] = \texttt{B}$
  - $P_3 = A[2, 2] = \texttt{C}$
  - $P_4 = A[3, 2] = \varepsilon$
- "BAC", with $l = 0, r = 1, k = 1$, since:
  - $P_1 = A[0, -1] = \varepsilon$
  - $P_2 = A[0, 0] = \texttt{A}$
  - $P_3 = A[1, 1] = \texttt{B}$
  - $P_4 = A[2, 2] = \texttt{C}$

D. Delivering Orders

1 second, 256 megabytes

Iván has just embarked in his newest entrepreneurship: a charbroiled burgers stand. The unique seasoning that he uses has attracted a lot of customers, so he's figuring out new ways to expand his business.

Since he has really loyal customers he knows that each day he will receive N burger orders, always being requested in the same order (sequence) as in previous days. Every order asks for certain quantities of each of the K ingredients that are available in the stand. Iván can only fulfill an order if, for each ingredient, he has an available amount **greater than or equal to** the amount that is being requested.

To minimize costs Iván wants to start buying items in bulk, for this he will start having a large inventory for each of the K ingredients. As he has bought his first batch in bulk, he would like to know how much time will they last.

Help Iván find out when will he run out of his current inventory and won't be able to fulfill another one. Hurry up, a hoard of hungry customers is waiting!

Input
On the first line, an integer $K - (1 \leq k \leq 100)$ – the number of different ingredients.

On the second line, K integers a_i $(0 \leq a_i \leq 10^{12})$ – the available quantity for the $i-$th ingredient.

On the third line, an integer $N - (1 \leq N \leq 10^4)$ – the number of orders that will arrive each day.

The following N lines will have $K - (0 \leq b_{ij} \leq 10^{12})$ numbers each, for the $i-$th line the $j-$th number will indicate the quantity of the ingredient of type j that the i-th order needs. It is guaranteed that each order will need at least one ingredient.

Output
Two numbers, indicating the first day Iván can't fulfill all the orders, and the index (1 based) of the first order Iván couldn't fulfill.

input

```
4
7 20 11 10
3
0 2 1 1
1 3 2 1
1 2 1 2
```

output

```
3 3
```

input

```
4
3 7 2 4
2
1 2 3 1
3 2 2 1
```

output

```
1 1
```

E. Elisas Boxes

1 second, 256 megabytes

In the quaint village of BoxTown, Elisa, the clever organizer, has been tasked with arranging a shipment of magical artifacts. There are N enchanted boxes, each with its own capacity. Elisa's goal is to fit M artifacts into one of these boxes, ensuring the safety and organization of the village's treasures. Each box i can hold up to a_i artifacts. Elisa needs to find a box that can accommodate all M artifacts. If multiple boxes can fit the artifacts, Elisa prefers the box with the smallest index to keep things tidy.

Input
The first line contains two integers N and M $(1 \leq N \leq 10^5$, $1 \leq M \leq 10^5)$, representing the number of boxes and the number of artifacts to store, respectively. The second line contains N integers $a_1, a_2, \ldots, a_N$ $(1 \leq a_i \leq 10^5)$, where a_i represents the capacity of the box with index i.

Output
Output a single integer representing the 1-based index of the box that can store all M artifacts. If no such box exists, output -1. If multiple boxes can store the artifacts, output the index of the box with the smallest index that can store the artifacts.

input

```
5 10
15 20 10 12 10
```

output

```
1
```

input

```
5 100
15 20 10 12 10
```

output

```
-1
```

F. First Problem

10 seconds, 1024 megabytes

Germán is new to writing programming problems. To get started, he decided to create a simple one (this is the very first problem he has written).

You are given a list of numbers B of length N, and K operations that can be of any of the following three types:

- `Q l r` — Query the maximum number in the range $[l, r]$ of the list B.
- `A l r` — Add one to every number in the range $[l, r]$ of the list B.
- `R l r` — In the range $[l, r]$ of the list B, reset to 0 every number that is equal to the global maximum value of the entire list B (not just the maximum in $[l, r]$).

Your task is to process all the given operations.

Input

The first line of input contains two integers N and K ($1 \le N, K \le 10^5$), the size of B and the number of operations respectively.

In the second line N integer numbers B_i ($1 \le B_i \le 10000$) separated by a space.

Each of the next K lines contains a single operation in the form: C l r, where C is a character denoting the type of operation (`Q`, `A`, or `R`), and l and r ($1 \le l \le r \le N$) define the range on which the operation is to be performed.

Output

For every operation of type `Q`, print a line with the result of the query.

input

```
10 10
1 2 3 4 5 6 7 8 9 10
Q 1 10
R 1 10
A 1 10
Q 1 10
R 1 7
Q 5 10
R 7 10
Q 6 10
A 6 10
Q 1 6
```

output

```
10
10
10
9
8
```

G. Generating Polygons

1 second, 256 megabytes

Geosé is preparing a problem for the upcoming ICPC Grand Prix of Solidland competition. He is busy writing the model solution and needs your help generating the tests.

The test should consist of a **simple** polygon with n vertices of **integer coordinates** and area equal to A. Recall that a polygon is simple if it has no holes and its boundary does not intersect itself. Additionally, the polygon **should not** contain any internal angle equal to 180°.

Build a polygon that satisfies the required constraints, or determine that it does not exist.

Input

Each test contains multiple test cases. The first line contains the number of test cases ($1 \le t \le 1,666$). The description of the test cases follows.

The first line of each test case contains two integers n and A ($3 \le n \le 5000$ and $1 \le A \le 10^8$).

It is guaranteed that the sum of n over all test cases does not exceed 5000.

Output

For each test case, print "No" if such a Polygon can not be built. Otherwise, print "Yes" followed by n lines with the coordinates of the vertices of the polygon in clockwise or counterclockwise order. The coordinates of the vertices should be integers with absolute value not exceeding 10^9.

If there are multiple solutions, print any of them.

input

```
4
4 100
5 10
6 30
10 1
```

output

```
Yes
0 0
0 10
10 10
10 0
Yes
0 0
2 0
3 2
1 4
-1 2
Yes
1 0
4 0
6 3
4 6
1 6
-1 3
No
```

H. Huron Designs

1 second, 256 megabytes

Tony works in Huron Designs, a website that receives requests for advertising design projects. Right now, Tony has n open requests, where the deadline of the i-th request is d_i seconds from now, the base profit of it is p_i, and the time to complete it is c_i after Tony starts working on it. Tony can work on at most one request at the same time.

A feature of Huron Designs is that the clients can give an optional bonus to the designers to recognize their work. You, as the assistant of Tony, have analyzed the previous transactions of the clients, and you have estimated that the client of the i-th request would pay x_i of bonus if Tony completes the i-th request on or before y_i seconds from now. You don't know the exact value of x_i and y_i, but you know that x_i is a real number chosen uniformly at random from the range $[lx_i, rx_i]$ and y_i is a real number chosen uniformly at random from the range $[ly_i, ry_i]$.

Your task is to find a subset of the requests that Tony should accept and the order in which the chosen requests should be completed to maximize the expected value of Tony's profit. Each one of the chosen requests should be completed on or before its respective deadline.

Input

The first line contains an integer n ($1 \le n \le 20$).

Each of the following n lines contains seven integers $d_i, p_i, c_i, lx_i, rx_i, ly_i, ry_i$ ($1 \le d_i, p_i, c_i \le 10^9$, $0 \le lx_i \le rx_i \le 10^9$ and $1 \le ly_i \le ry_i \le 10^9$).

Output

Print one number — the maximum expected value of Tony's profit.

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

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

input

```
2
5 10 5 2 2 1 4
15 20 5 3 3 10 15
```

output

```
33.000000000000
```

input

```
2
10 15 8 50 50 1 8
20 1 7 20 30 6 10
```

output

```
19.750000000000
```

I. ICPC Challenge

1 second, 256 megabytes

German had always dreamed of joining the legendary tech company known as Infinite Compilation, Perpetual Confusion (ICPC). It was a company surrounded by mystery and challenge, where logic was king and only the brightest minds could find their place.

One day, German discovered something unusual about how the company worked. Every employee reported to exactly one other person, except for the CEO, who stood at the very top and had employee ID 1. From the CEO down, each person could have multiple people working under them, forming a giant organizational chart that branched out in every direction.

To join the company, German didn't need to submit a resume or pass a technical test. Instead, there was a unique challenge: he had to answer a series of questions about how the employees were organized and what they knew.

Each employee at ICPC was an expert in one specific topic. The questions German would face went something like this: "For a given employee x, how many people under their supervision—including themselves—know a specific topic k?"

His friend Sebastian, who had once tried to join ICPC, remembered these questions clearly. Before he was caught snooping through the company's systems, he managed to get a copy of the entire employee structure—who reports to whom—and the exact list of questions German would be asked.

Unfortunately, Sebastian was discovered before he could find and share the answers.

Now it's up to you to help German. With the structure and questions in hand, he just needs the correct answers so he can memorize them before his interview.

Will you help him prepare and finally enter the world of Infinite Compilation, Perpetual Confusion?

Input

The first line of input contains two integers, N $(1 \leq N \leq 100000)$ and Q $(1 \leq Q \leq 100000)$ — the number of employees and the number of queries, respectively.

The second line contains N integers K_i $(1 \leq K_i \leq 10^9)$, where K_i represents the topic known by employee i.

Each of the next $N-1$ lines contains two integers X and Y $(1 \leq X, Y \leq N,\ X \neq Y)$, meaning that employee Y reports directly to employee X.

Each of the next Q lines contains two integers X and k $(1 \leq X \leq N,\ 1 \leq k \leq 10^9)$, representing a question asking how many employees under the supervision of employee X (including X) know topic k.

Output

For every question, print a line with the correct answer.

input

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

output

```
2
1
1
4
```

J. JuPaels Palindrome

1 second, 1024 megabytes

In the mystical realm of Spellbound, a powerful wizard named JuPael has discovered a magical string that holds the key to creating powerful incantations. This string is composed of characters from the english alphabet. The wizards aim to find the largest segment of this string that can be rearranged to form a palindrome, a word that reads the same forwards and backwards.

Your task is to assist the wizards by determining the length of the longest contiguous segment of the string that can be rearranged into a palindrome.

Input

The first line contains an integer N $(1 \leq N \leq 10^6)$, representing the length of the string. The second line contains the string S, composed of lowercase letters from the english alphabet.

Output

Output a single integer representing the length of the largest segment that can be rearranged into a palindrome.

input

```
15
abacabadabacaba
```

output

```
15
```

input

```
5
abcde
```

output

```
1
```

input

```
10
aaaaaaaaaa
```

output

```
10
```

K. Killable Demon

1 second, 256 megabytes

Bausffs, known as the Killable Demon, was a fearless tactician who thrived on taking risks. He had a unique playstyle: he deliberately placed himself in difficult positions where every move could result in either great reward or crushing loss. His goal was simple — always come out with more gold than his opponent, no matter the odds.

But even Bausffs had limits.

That's where we come in. We are the Unkillable Demon eternal, analytical, and all-seeing. We know every decision that could be made in the game. We know their outcomes, the relation between them, and the true value they offer.

Each decision i is characterized by an integer:

$$D_i = (\text{Amount of Gold that Bausffs gains}) - (\text{Amount of Gold that Baus}$$

Some decisions depend on others. That is, there exist some pair (A, B) such that Bausffs cannot make decision A unless he has already made decision B. Decisions can only be made once and, if necessary, several decisions can be made simultaneously.

Bausffs, eager to win and outplay his enemy, comes to us for help.

"Tell me," he says, "what is the maximum gold advantage I can gain if I choose my decisions wisely?"

And we, the Unkillable Demon, must answer.

Input

The first line of input contains two integers N and M $\left(1 \leq N \leq 200, 1 \leq M \leq \frac{n(n-1)}{2}\right)$, representing the number of decisions and the number of relations, respectively.

The second line contain N integers D_i $(-1000 \leq D_i \leq 1000)$ separated by a space indicating the true value of that decision.

Each of the next M lines contains two integers separated by a space, u_i, v_i $(1 \leq u_i, v_i \leq N)$ indicating that taking decision u_i is a requirement for taking decision v_i.

Output

Print a single line — The maximum gold advantage the Killable Demon can obtain.

input

```
6 4
-10 -10 22 -3 5 1
1 3
2 3
3 4
4 5
```

output
5

L. La Vaca Saturno Saturnita vs Tung Tung Tung Sahur

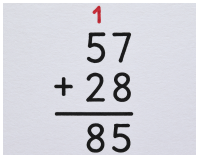
0.25 seconds, 1024 megabytes

La Vaca Saturno Saturnita and Tung Tung Tung Sahur will have a battle in the Base-b Game.

The Base-b Game is played with a collection of piles of stones and an integer b chosen before the game starts. The two players of the game take turns. On each turn, a player selects a pile of stones and adds a **positive** number of stones such that the following two conditions are satisfied:

1. Let a be the current size of the pile and x the stones added. The number of digits of x in base b doesn't exceed the number of digits of a in base b. For example, 11 has 2 digits in base 10, 9 has 4 digits in base 2, and 0 has 1 digit in any base.
2. The sum $a + x$ doesn't produce any carry when the sum is performed with the basic addition method in base b.

For example, if $b = 10$, a player can't add 28 stones to a pile of size 57 since the sum $57 + 28$ produces a carry from the units to the tens when performing the sum in base 10, as shown in the image below. On the other hand, if $b = 100$, a player can add 28 stones to a pile of size 57.



The player who can't make a move loses the game. If the game continues for more than $2025!^{2025!}$ turns, it is declared a tie.

La Vaca Saturno Saturnita will take the first turn. Tung Tung Tung Sahur considered that this was unfair, so he was allowed to choose the sizes of the piles. Sahur can choose any subset (possibly empty) of $\{l, l+1, l+2, \ldots, r\}$ as the sizes of the piles. Note that Sahur can't take multiple piles of the same size.

Tung Tung Tung Sahur is not good at strategy games, so he asks for your help to count the number of ways to choose the sizes of the piles such that he (the second player) has a winning strategy if both of them play optimally. Since the answer can be huge, find it modulo 998244353. The order of sizes does not matter, only the chosen subset.

Input

The first line contains three integers b, l and r ($2 \le b \le 100$ and $0 \le l \le r < 10^{100}$).

Output

Print the anwser modulo 998244353.

input
10 348 348
output
1

input
2 12345678901234567890 98765432109876543210
output
520579205