

# 2020 “Lenovo Cup” USST Campus Online Invitational Contest

## Official Problem Set

University of Shanghai for Science and Technology

May 30, 2020



上海理工大学  
UNIVERSITY OF SHANGHAI FOR SCIENCE AND TECHNOLOGY

公共实验中心  
Public Experiment Center



上海理工大学  
UNIVERSITY OF SHANGHAI FOR SCIENCE AND TECHNOLOGY

教务处  
Academic Affairs Office



上海理工大学  
UNIVERSITY OF SHANGHAI FOR SCIENCE AND TECHNOLOGY

创新创业学院  
Innovative and Entrepreneurship Institute

Lenovo 联想

### Important Note:

- If you want to get prize from the contest sponsor, please complete the required information (real name, school and e-mail) in EOJ personal settings.

# PROBLEMS

A. Archmage	1
B. Bamboo Leaf Rhapsody	3
C. Cheat Sheet	5
D. Disaster Recovery	6
E. Eight Digital Games	8
F. Fake Algorithm	11
G. Gentle Jena	13
H. Hay Mower	15
I. Immortal Trees	17
J. JXC!!	20
K. K-Shift Array	23
L. Lottery Tickets	25
M. MITE	26

**Problem Setter:** Xuancheng Jin, Haoyun Ren

**Coach:** Chunqiao Zhou

**Sponsor:** Lenovo

**Special thanks to EOJ developers for wonderful platform EOJ, Jiadong Xie for patient coordination and Sixiao Gao for English consultant.**

## Problem A. Archmage

Input file: Standard Input

Time limit: 2 seconds

Output file: Standard Output

Memory limit: 256 megabytes

Archmage (AM for short) is a hero unit in Watercraft III.

He has  $n$  mana **at most** and two powerful skills which enable him to show great strength in battle:

1. Use  $x$  mana to summon a Water Element.
2. Brilliance Aura restores  $y$  mana for AM in a second, and it is the **only** way to restore mana. Since his mana cannot exceed the upper limit, the superfluous mana will dissipate. i.e., suppose you have  $p$  mana before the restoration, your mana will become  $\min(p + y, n)$ .

Since the power of an Archmage is tremendous, the upper limit of mana  $n$  is always greater than or equal to  $x + y$ .

Every second, Archmage will summon **exactly one** Water Element (if the mana is enough, i.e., his mana won't be less than 0 after that) or do nothing. Then no matter what he did before, he will restore  $y$  mana.

Archmage has  $n$  mana at the end of the second 0, and the game starts at the beginning of the second 1.

He wants to know how many Water Elements he can summon at most before the end of the second  $m$ .

### Input

The input consists of multiple test cases.

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^5$ ), indicating the number of test cases.

Each of the next  $t$  lines contains 4 integers  $n, m, x, y$  ( $1 \leq m, x, y \leq 10^9, x + y \leq n \leq 2 \times 10^9$ ).

### Output

For each test case, output the answer in a line.

### Samples

Standard Input	Standard Output
3	2
2 2 1 1	3
4 4 2 1	6
6 10 4 2	

### Note

In test case 1, Archmage can cast spells every second, so the answer is 2.

In test case 2, here's a way for Archmage to cast spells 3 times.

Second 1: Archmage cast spells, and there is  $4 - x + y = 3$  mana left.

Second 2: Archmage cast spells, and there is  $3 - x + y = 2$  mana left.

Second 3: Archmage cast spells, and there is  $2 - x + y = 1$  mana left.

Second 4: Archmage doesn't have enough mana so he can do nothing, and there is  $1 + y = 2$  mana left.

## Problem B. Bamboo Leaf Rhapsody

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

*Do you know who will fulfill people's wishes on Tanabata?*

*It's  $\alpha$  Lyra and  $\alpha$  Aquila.*

*By the way, Earth is 25 light-years away from  $\alpha$  Lyra and 16 light-years away from  $\alpha$  Aquila. That is to say, it will take 25 or 16 years for the messages sent from Earth to reach these two constellations. That's for granted. Do you understand?*

—*The Boredom of Haruhi Suzumiya, 2004*

This is the 16th year since the publication of the novel. Apart from the short stories, the author has not updated the series for nine years.

Setsuna is a big fan of the novel. She makes a wish that the author would update as soon as possible, and she will send this message to the stars.

The universe can be seen as a three-dimensional rectangular coordinate system in which the earth is the origin(i.e., the coordinate of the earth is  $(0, 0, 0)$ ).

Suppose there are  $n$  stars, and the coordinate of the  $i$ -th one is  $(x_i, y_i, z_i)$ . Any one of them can receive her message.

The speed of the message is exactly 1 unit per year in the universe, and the direction can be arbitrary.

Setsuna wants to know the shortest time for the message to reach at least one of the stars after she sends it.

You can assume that Einstein's theory of relativity doesn't work in Setsuna's world.

### Input

The first line contains one integer  $n$  ( $1 \leq n \leq 1000$ ), indicating the number of the stars.

The  $i$ -th of the next  $n$  lines contains three integers  $x_i, y_i, z_i$  ( $-1000 \leq x_i, y_i, z_i \leq 1000$ ), indicating the coordinate of the  $i$ -th star.

### Output

Output the shortest time in years, the value should be rounded to three decimal places.

### Samples

Standard Input	Standard Output
3 0 1 1 2 0 0 0 -2 0	1.414

## Note

The first star can receive the message in  $\sqrt{2}$  years.

The second and the third star can receive the message in 2 years.

So the answer is  $\sqrt{2}$  years, and it should be rounded to 1.414 .

## Problem C. Cheat Sheet

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

University of Shanghai for Science and Technology starts a course called **Film Appreciation of *Black Album*** recently. To be the best "Blackologist" in the university, Setsuna is actively preparing for the exam.

The examination of the course is open book; that is to say, you can only take one single-sided cheat sheet to the exam. The cheat sheet can write  $n$  characters at most.

Setsuna has  $m$  keywords that she wants to write on the cheat sheet. Her memory is not very good, so there may be some duplicate keywords. Each keyword consists of several visible characters (visible characters refer to characters with ASCII code between 33 and 126 inclusive).

For both readability and neatness, keywords written on the cheat sheet should be separated by at least one space and must be different from each other.

Setsuna wants to know how many **distinct** keywords she can write down on the cheat sheet at most.

Uppercase and lowercase letters are considered different characters.

### Input

The first line contains two integers  $n, m$  ( $1 \leq n, m \leq 1000$ ).

The second line contains  $m$  keywords separated by exactly one space. The length of each keyword is no more than 100. It is guaranteed that keyword only consists of visible characters.

### Output

Output one integer indicating the answer.

### Samples

Standard Input	Standard Output
40 5 myworld lusto KR12138 oneman233 SetsunaQAQ	4
7 2 ^ ^ ^ ^ _ _ ^ _	1

### Note

In sample 1, it takes 42 characters to write all the words down. So Setsuna can write down at most four.

In sample 2, there is only one keyword.

## Problem D. Disaster Recovery

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

On the magical land Byteland, there is an ancient kingdom called Fibonacci Kingdom.

There are  $n$  cities in the kingdom (numbered from 1 to  $n$ ), and  $\text{Fib}_i$  residents in the  $i$ -th city.

$\text{Fib}_i$  denotes the  $i$ -th number of **Fibonacci sequence**, such that each number is the sum of the two preceding ones, starting from 1 and 1. That is,  $\text{Fib}_0 = 1, \text{Fib}_1 = 1$  and  $\text{Fib}_n = \text{Fib}_{n-1} + \text{Fib}_{n-2}$  for  $n > 1$ .

The beginning of the sequence is thus: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,  $\dots$ .

Unfortunately, one day, an earthquake occurred.

The earthquake destroyed the whole kingdom's transportation system.

To deliver disaster relief supplies, the king decided to reconstruct the road.

$m$  plans have been proposed, and the  $i$ -th plan is to reconstruct the **bidirectional** road linking  $x_i$  and  $y_i$ , denoted as  $(x_i, y_i)$ .

The magic is that it takes exactly  $\text{Fib}_{x_i} + \text{Fib}_{y_i}$  bitcoins to reconstruct the road  $(x_i, y_i)$ .

The king wants to choose some plans to implement, but the rules below must be followed:

1. After the reconstruction, the cities are mutually reachable (i.e., there is at least one path connecting any two cities).
2. Minimize the bitcoin consumption.
3. In the premise of the least bitcoin consumption, the largest degree  $D$  among all the cities should be minimized (suppose  $d_i$  is the degree of  $i$ -th city,  $D = \max_{1 \leq i \leq n} d_i$ ), because the residents of the kingdom are not good at looking for ways. The definition of the degree of a city is the number of roads **directly** connected to the city.

Setsuna, the savior of the kingdom, has easily calculated the minimum consumption of bitcoin, but she desperately wants to know the minimum  $D$  she can get after reconstruction.

It's guaranteed that the answer always exists.

### Input

The first line contains two integers  $n, m$  ( $2 \leq n \leq 10^5, n - 1 \leq m \leq \min(\frac{n \times (n-1)}{2}, 2 \times 10^5)$ ).

The  $i$ -th of the next  $m$  lines contains two integers  $x_i, y_i$  ( $1 \leq x_i, y_i \leq n, x_i \neq y_i$ ).

### Output

Output one integer which indicates the minimum  $D$  Setsuna can get.

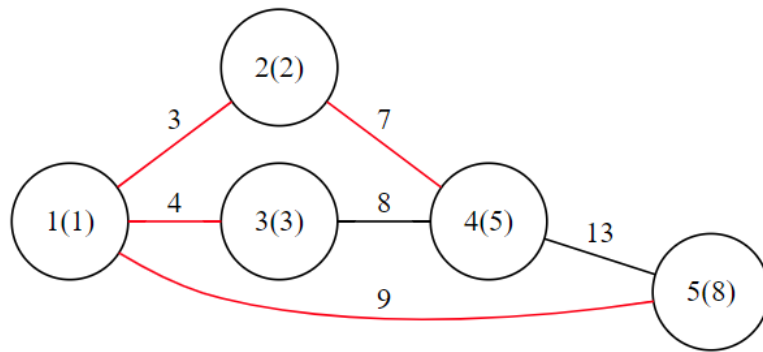


## Samples

Standard Input	Standard Output
5 6 2 1 2 4 1 3 3 4 4 5 1 5	3

## Note

The graph of the sample is as follows.



In this situation, bitcoin consumption is  $3 + 4 + 7 + 9 = 23$ . The degree of city 1 is 3; the degree of city 2 is 2; the degree of city 3, 4, 5 is 1.

It can be proved that such solution is the best, so the answer is 3.

## Problem E. Eight Digital Games

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

Setsuna has been obsessed with an electronic video game called "Eight Digital". It is a puzzle game which is too difficult for Setsuna, so she has been stuck in the first level for a long time.

In the game, you have a string of length  $n$  containing only digits from 1 to 8. Let's call this string  $S$ ,  $S_i$  denotes the  $i$ -th character of  $S$ .

At the end of the game, the game system will calculate the penalties of inversions, and will deduct your coins. Inversion is a pair  $(i, j)$  which satisfies both  $i < j$  and  $S_i > S_j$ . The game system will deduct you  $P_{S_i, S_j}$  coins for each inversion  $(i, j)$ .

For example, string 85511 will cost you  $2 \times P_{8,5} + 2 \times P_{8,1} + 4 \times P_{5,1}$  coins, and string 1234 will cost you 0 coins.

Before the end of the game, you can do **arbitrary** times of the operation. Each operation is to select two different numbers  $x$  and  $y$  ( $1 \leq x, y \leq 8$ ), then all  $x$  in the string will become  $y$ , and all  $y$  will become  $x$  and the game system will deduct you  $C_{x,y}$  coins.

For example, you can spend  $C_{1,3}$  to convert string 131233 into string 313211.

To help poor girl Setsuna, you are required to find the minimum cost of coins to finish the game.

### Input

The first line contains one integer  $n$  ( $1 \leq n \leq 10^5$ ).

The second line contains a string  $S$ . It is guaranteed that the length of  $S$  is  $n$  and  $S$  only contains digits from 1 to 8.

The next 8 lines contains 8 integers each, where the  $j$ -th integer of the  $i$ -th line is  $P_{i,j}$  ( $0 \leq P_{i,j} \leq 10^7$ ). It is guaranteed that  $P_{i,j} = 0$  holds for  $i \leq j$ .

The next 8 lines contains 8 integers each, where the  $j$ -th integer of the  $i$ -th line is  $C_{i,j}$  ( $0 \leq C_{i,j} \leq 10^{12}$ ). It is guaranteed that  $C_{i,i} = 0$  and  $C_{i,j} = C_{j,i}$ .

### Output

Output one integer indicating the minimum cost.

**Samples**

Standard Input	Standard Output
<pre> 5 54321 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 </pre>	<pre> 2 </pre>
<pre> 6 222112 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 2 2 2 2 2 2 1 0 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2 2 0 </pre>	<pre> 4 </pre>

**Note**

In sample 1, you can spend 1 coin to convert 54321 into 14325. Then, spend another 1 coin to convert it into 12345.

In sample 2, you can spend 2 coins to convert 222112 into 222882, then end the game. The inversions (4,6) and (5,6) will cost you  $2 \times P_{8,2} = 2$  coins. So the total cost is 4 coins.

## Problem F. Fake Algorithm

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

Setsuna has been fascinated by coprime pairs recently, so she decided to play a game about numbers.

Here are  $n$  integers  $a_1, a_2, \dots, a_n$  ( $2 \leq a_i \leq 10^{18}$ ). Setsuna wants to divide the numbers into as few groups as possible so that each number is exactly in one group and the numbers in each group are coprime with each other.

We say that positive integers  $x$  and  $y$  are coprime if and only if  $\gcd(x, y) = 1$ , where  $\gcd(x, y)$  refers to their greatest common divisor.

However, Setsuna is not good at math, so she only comes up with a fake algorithm.

---

### Algorithm 1 Setsuna's Algorithm

---

**Input:**  $A : a_1, a_2, \dots, a_n$ ;  $n$  : size of  $A$

**Output:** Minimum number of groups

```

1: function CHECK( $x, y, G, A$ ) // Check whether  $a_y$  and  $a_x$  can be in the same group.
2:    $result \leftarrow \text{True}$ 
3:   for  $i = 1 \rightarrow n$  do
4:     if  $\gcd(a_i, a_y) \neq 1$  and  $G_i = G_x$  then
5:        $result \leftarrow \text{False}$ 
6:     end if
7:   end for
8:   return  $result$ 
9: end function
10:
11: function SOLVE( $A, n$ )
12:   //  $G_i$  indicates which group is  $a_i$  divided into, and  $cnt$  indicates the number of groups.
13:   Initial  $cnt \leftarrow 0$  and fill  $G$  with 0
14:   for  $i = 1 \rightarrow n$  do
15:     if  $G_i \neq 0$  then
16:       Continue
17:     end if
18:      $cnt \leftarrow cnt + 1$ 
19:      $G_i \leftarrow cnt$ 
20:     for  $j = i + 1 \rightarrow n$  do
21:       if  $G_j = 0$  and CHECK( $i, j, G, A$ ) then
22:          $G_j \leftarrow cnt$ 
23:       end if
24:     end for
25:   end for
26:   return  $cnt$ 
27: end function

```

---

Her greedy algorithm is obviously wrong, so we need to hack her solution.

You are given one integer  $k$ . You need to find the input consisting of  $n$  ( $n \leq 300$ ) integers and a corresponding partition to the input (**not necessarily** be the minimum partition), which satisfies  $X = Y + k$ , where  $Y$  is the number of groups of your partition and  $X$  is the answer of her algorithm to your input.

It can be proved that the answer always exists.

## Input

The input contains one integer  $k(1 \leq k \leq 7)$ .

## Output

In the first line, output two integers  $n, Y(1 \leq Y \leq n \leq 300)$ .

In the second line, output  $n$  integers  $a_1, a_2, \dots, a_n(2 \leq a_i \leq 10^{18})$ , separated by spaces.

In the third line, output  $n$  integers  $G_1, G_2, \dots, G_n(1 \leq G_i \leq Y)$ , separated by spaces, where  $g_i$  indicates which group is  $a_i$  divided into.

If there are multiple solutions, you can output any.

## Samples

Standard Input	Standard Output
1	4 2 8 3 45 100 1 2 1 2

## Note

In the sample, Setsuna will get 3 groups:  $\{8, 3\}, \{45\}, \{100\}$ , but we have a better partition:  $\{8, 45\}, \{3, 100\}$ .

## Problem G. Gentle Jena

Input file: Standard Input

Time limit: 2 seconds

Output file: Standard Output

Memory limit: 512 megabytes

... *Why don't you come to the planetarium?*

*The beautiful twinkling of eternity that will never fade, no matter what.*

*All the stars in the sky are waiting for you...*

A planetarium, it was once where people went to soothe their hearts by gazing at a starry sky.

And Yumemi, a planetarium guide who has been waiting for a visitor for a long time, is looking up at the stars.

Stars in the sky can be defined by their brightness, denoted by  $S = \{b_1, b_2, \dots, b_n\}$ .

Yumemi found that stars always appear in groups, and she thinks the beauty of the starry night depends on the darkest one, so she defined the beauty value as

$$B(S) = \sum_{1 \leq l \leq r \leq |S|} f(l, r) \bmod 998244353$$

where

$$f(l, r) = \min_{l \leq i \leq r} \{b_i\}$$

But the night sky is not always the same. The movement of celestial bodies will make more and more stars in the night sky. The following events will happen in order every second:

1. There are exactly  $i$  stars in the sky so  $S = \{b_1, b_2, \dots, b_i\}$ . A star with brightness  $b_{i+1}$  appears, and it will be appended to the end of the sequence  $S$ , so it will be  $S = \{b_1, b_2, \dots, b_{i+1}\}$ .
2. Yumemi records the value (i.e., She calculates the value of  $B(S)$ ).

At the beginning, there is no star in the sky, so  $S = \{\}$  initially.

As time goes by, because she is not as good at calculation as before, the task will be given to you.

**Note that the sequence is given in a modified way.**

### Input

To avoid huge input, we use Linear Congruential Method to generate input data, and your solution should be **on-line**.

The first line contains six integers  $n, p, x, y, z, b_1$  ( $1 \leq n \leq 10^7, 2 \leq p \leq 10^9, 0 \leq x, y, z, b_1 < p$ ), indicating the number of the stars and the parameters of the data generator.

You need to generate the sequence  $\{b_1, b_2, \dots, b_n\}$  by yourself using the following formula:

$$b_{i+1} = (x \times A_i + y \times b_i + z) \bmod p$$

where  $A_i$  is the value of  $B(S)$  when  $|S| = i$ .

## Output

To avoid huge output, you only need to output  $\bigoplus_{1 \leq i \leq n} A_i$ , where " $\bigoplus$ " denotes the bitwise XOR operation ([https://en.wikipedia.org/wiki/Bitwise\\_operation#XOR](https://en.wikipedia.org/wiki/Bitwise_operation#XOR)).

## Samples

Standard Input	Standard Output
5 13 2 0 1 3	27

## Note

In the first second,  $S = \{b_1\} = \{3\}$ , so  $A_1 = f(1, 1) \bmod 998244353 = 3$ .

It can be inferred that  $b_2 = (2A_1 + 1) \bmod 13 = 7$ .

In the second second,  $S = \{b_1, b_2\} = \{3, 7\}$ , so  $A_2 = [f(1, 1) + f(2, 2) + f(1, 2)] \bmod 998244353 = 13$ .

It can be inferred that  $b_3 = (2A_2 + 1) \bmod 13 = 1$ .

Keep calculating, and you will get  $A_3 = 16, b_4 = 7, A_4 = 26, b_5 = 1, A_5 = 31$ .

So you should output  $A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 = 27$ .



## Problem H. Hay Mower

Input file: Standard Input

Time limit: 2 seconds

Output file: Standard Output

Memory limit: 256 megabytes

Are you tired of city life? Have you ever had illusions of pastoral peace? The clean atmosphere, the closeness to nature and the gentle pace of living, all made Setsuna yearn for the pastoral life more.

In order to experience the simple pastoral life, Setsuna has moved to Star Valley and started her farming journey.

Soon, she discovers the problem: overgrown weeds are harming her farm. In Chinese, we call it "Sheng Cao". She realized that weeding should be put at the top priority.



The farm can be described as an  $n \times m$  matrix. The growth rate of weed in row  $i$  and column  $j$  is denoted as  $a_{i,j}$ , indicating that the weed will grow  $a_{i,j}$  units every **beginning** of the moment. At the end of moment 0, there is no weed on the farm.

Setsuna will use mower  $k$  times, where the  $i$ -th use occurs at the **end** of moment  $t_i$ . Each use of the mower completely removes weeds in a row or column.

Setsuna wonders how many units of weed she will remove.

The answer might be very large, so please output the desired answer modulo 998244353.

### Input

The first line contains three integers  $n, m, k$  ( $1 \leq n, m \leq 500, 1 \leq k \leq 3 \times 10^5$ ).

The next  $n$  lines contains  $m$  integers each, where the  $j$ -th integer of the  $i$ -th line is  $a_{i,j}$  ( $0 \leq a_{i,j} \leq 10^{18}$ ).

The  $i$ -th of the next  $k$  lines contains one character and two integers.

- r  $x_i t_i$  - Setsuna clears the weeds in row  $x_i$  at the end of moment  $t_i$ .
- c  $y_i t_i$  - Setsuna clears the weeds in column  $y_i$  at the end of moment  $t_i$ .

It is guaranteed that  $1 \leq x_i \leq n, 1 \leq y_i \leq m, 1 \leq t_i \leq 10^{18}$  hold for  $1 \leq i \leq k$  and  $t_i$  is strictly increasing.

### Output

Output one integer indicating the answer modulo 998244353.

**Samples**

Standard Input	Standard Output
2 2 3 1 2 3 4 r 1 5 c 2 6 r 1 7	45
3 4 1 1 2 3 4 5 6 7 8 9 10 11 12 r 1 10000000000000000000	172998509

**Note**

Sample 1:

At the end of moment 0, the farm looks like

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

At the end of moment 5, Setsuna has cleared row 1 and 15 units of weed has been cut, so the farm looks like

$$\begin{bmatrix} 0 & 0 \\ 15 & 20 \end{bmatrix}$$

At the end of moment 6, Setsuna has cleared column 2 and 26 units of weed has been cut, so the farm looks like

$$\begin{bmatrix} 1 & 0 \\ 18 & 0 \end{bmatrix}$$

At the end of moment 7, Setsuna has cleared row 1 and 4 units of weed has been cut, so the farm looks like

$$\begin{bmatrix} 0 & 0 \\ 21 & 4 \end{bmatrix}$$

So the answer is  $15 + 26 + 4 = 45$  units.

---

## Problem I. Immortal Trees

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

The beautiful campus of University of Shanghai for Science and Technology is lined with trees. It takes ten years to grow trees and a hundred years to nurture talents. The trees have witnessed the growth of generations. Unconsciously, the season of graduation has quietly arrived.



To commemorate the contributions of seniors to the algorithm contest of University of Shanghai for Science and Technology, everyone decides to plant an unrooted tree of  $n$  vertices in the school. Vertices are numbered from 1 to  $n$ .

It's difficult to satisfy everyone's aesthetic, so the tree should meet some constraints.

There are  $m$  constraints of type 1 and  $k$  constraints of type 2.

The  $i$ -th constraint of type 1 can be represented by a pair  $(x_i, y_i)$ , which indicates there should be an edge connecting vertex  $x_i$  and vertex  $y_i$ .

The  $i$ -th constraint of type 2 can be represented by a triple  $(op_i, x_i, deg_i)$ .  $op_i$  indicates the type of the restriction, 1 for the upper limit and 0 for the lower limit.  $x_i$  indicates the number of the vertex.  $deg_i$  indicates the extreme value of degree of vertex  $x_i$ .

For example, the constraint triple  $(1, 3, 1)$  means that the degree of vertex 3 should be **at most** 1, and the constraint triple  $(0, 1, 2)$  means that the degree of vertex 1 should be **at least** 2.

You need to count how many kinds of trees meet all constraints simultaneously. Notice that the answer might be very large, so please output the desired answer modulo 998244353.

There are some definitions you may need to know: an undirected connected graph is a tree if and only if it has  $n$  vertices and  $n - 1$  edges; two trees are considered different if and only if their adjacency matrixes are different.

## Input

The first line contains three integers  $n, m, k$  ( $2 \leq n \leq 60, 0 \leq m \leq n - 1, 0 \leq k \leq 60$ ).

The  $i$ -th of the next  $m$  lines contains two integers  $x_i, y_i$  ( $1 \leq x_i, y_i \leq n, x_i \neq y_i$ ).

The  $i$ -th of the next  $k$  lines contains three integers  $op_i, x_i, deg_i$  ( $0 \leq op_i \leq 1, 1 \leq x_i \leq n, 0 \leq deg_i < n$ ).

## Output

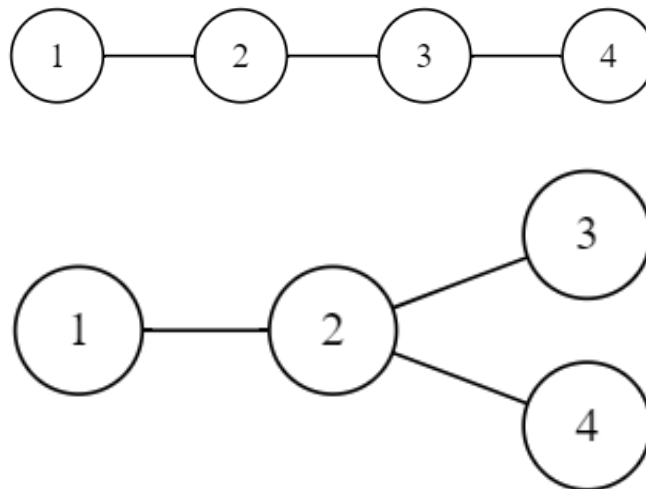
Output one integer in a single line.

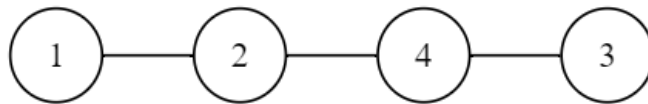
## Samples

Standard Input	Standard Output
4 3 1 1 2 2 1 1 2 1 1 1	3
4 3 1 1 2 2 3 1 4 0 1 2	1
4 3 0 1 2 2 3 3 1	0

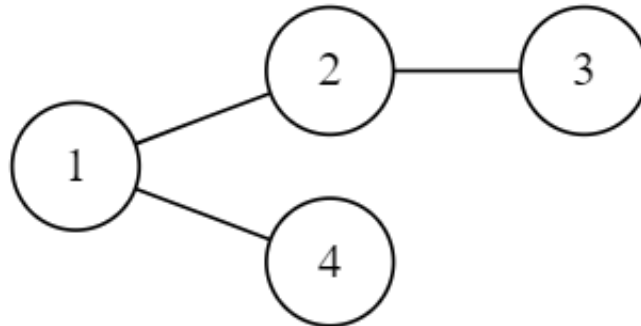
## Note

In sample 1, there are 3 trees that meet all constraints.





In sample 2, there is only 1 tree that meets all constraints.



In sample 3, there is no tree that simultaneously meets all constraints.

## Problem J. JXC!!

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

**This is an interactive problem.**

As we all know, JXC is the red sun of USST. Now, he is going to play an easy game with Setsuna.

Setsuna is given a grid  $n \times m$ . Rows are enumerated from 1 to  $n$  from up to down, and columns are enumerated from 1 to  $m$  from left to right. Cell, standing on the intersection of row  $x$  and column  $y$ , is denoted by  $(x, y)$ , and the number in cell  $(x, y)$  is denoted as  $A_{x,y}$ .

Every cell contains 0 or 1, but only JXC knows what they are.

Setsuna wants to know numbers in all cells of the grid. To do so we can ask the following questions:

- “?  $x_1 y_1 x_2 y_2$ ”, where  $1 \leq x_1 < x_2 \leq n, 1 \leq y_1 < y_2 \leq m$ . Setsuna can get the number of **good pairs** in the submatrix with  $(x_1, y_1)$  as the upper left corner and  $(x_2, y_2)$  as the lower right corner through this query. Obviously, the size of the submatrix she will output is not smaller than  $2 \times 2$ . **Good pair** is the adjacent pair with the same value. Formally, it is an unordered pair  $((x_1, y_1), (x_2, y_2))$  which satisfies both  $|x_1 - x_2| + |y_1 - y_2| = 1$  and  $A_{x_1, y_1} = A_{x_2, y_2}$ .
- “?  $x y$ ”, where  $1 \leq x \leq n, 1 \leq y \leq m$ . Setsuna can get the number in cell  $(x, y)$ . **She can perform no more than 5 queries of this type.**

For example, for the graph below, answer for “? 1 1 3 3” is 4 and answer for “? 3 4” is 0.

0	1	1	0
1	0	0	1
0	0	1	0

The diagram shows a 3x4 grid of cells. A red rectangle highlights a 3x3 submatrix from (1,1) to (3,3). Red lines connect adjacent cells with the same value: (1,2)-(1,3), (2,1)-(2,2), (2,2)-(2,3), (3,1)-(3,2), and (3,2)-(3,3). The cell (3,4) containing the value 0 is circled in red.

You are required to help Setsuna to determine all cells of the grid by asking not more than 2000 questions(including two kinds of questions and the guess of the answer). It can be shown that the answer always exists.

## Input

The first line contains two integers  $n, m (3 \leq n, m \leq 32)$ .

## Interaction Protocol

You begin the interaction by reading  $n, m$ .

To ask a question about submatrix  $(x_1, y_1), (x_2, y_2)$ , output “?  $x_1 y_1 x_2 y_2$ ” in a separate line.

Numbers in the query have to satisfy  $1 \leq x_1 < x_2 \leq n$  and  $1 \leq y_1 < y_2 \leq m$ .

To ask a question about the number in cell  $(x, y)$ , output “?  $x y$ ” in a separate line.

Numbers in the query have to satisfy  $1 \leq x \leq n$  and  $1 \leq y \leq m$ .

Don’t forget to ‘flush’, to get the answer.

In response, you will receive the number of good pairs if you asked the submatrix or the number in the cell otherwise.

In case your query was invalid or you asked more than 2000 queries (including two kinds of questions and the guess of the answer) or you asked more than 5 queries of the second type, program would print  $-1$  and finish the interaction. You would receive **Wrong answer** verdict. Make sure to exit immediately to avoid getting other verdicts.

When you determine numbers in all cells, output “!”. There must be a line break(‘\n’) after “!”.

Then output  $n$  lines, the  $i$ -th of which is a string of length  $m$ , corresponding to numbers in the  $i$ -th row of the grid.

Note that you have only 1 chance to guess the answer, and after outputting the answer, your program must be terminated immediately without outputting anything.

After printing a query, do not forget to output end of line and flush the output. Otherwise, you will get **Idleness limit exceeded**. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

## Samples

Standard Input	Standard Output
3 3	? 1 1
1	? 1 2
1	? 1 3
1	? 2 1
0	? 2 2
0	? 2 2 3 3
4	? 2 1 3 3
5	!
	111
	000
	100

## Note

The interaction process in the sample is as follows.

1	1	1
0	0	0
1	0	0

Setsuna first used all the second type of queries to get the numbers in the first five cells.

Then she asked about the  $2 \times 2$  submatrix in the lower right corner. JXC told her there are 4 good pairs, so it is easy to infer that this submatrix is all 0.

Finally she asked about the  $2 \times 3$  submatrix at the bottom. JXC told her there are 5 good pairs, so it is easy to infer that the lower left corner is 1.



## Problem K. K-Shift Array

Input file: Standard Input

Time limit: 6 seconds

Output file: Standard Output

Memory limit: 256 megabytes

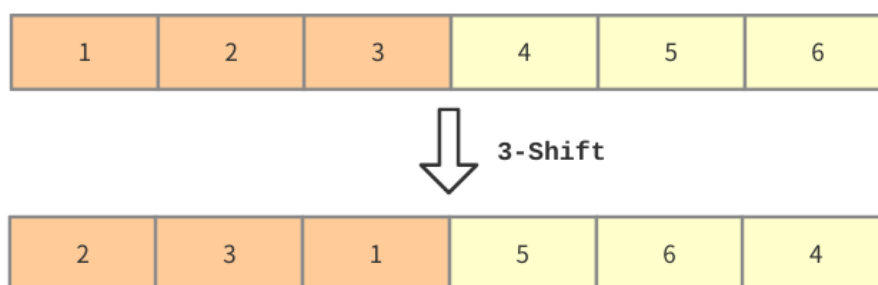
Setsuna, a lovely girl who likes to play with data structures, is going to share an interesting problem with you which is called  $K$ -Shift.

An array  $A$  can be  $K$ -Shifted if and only if the length of array  $A$  can be divided by  $K$  (i.e.,  $|A|$  must be a multiple of  $K$ ).

When Setsuna  $K$ -Shift the array  $A$ , the following events will happen in order:

1. From the beginning of  $A$ , every consecutive  $K$  elements are divided into a group, so there will be exactly  $\frac{|A|}{K}$  groups.
2. In every group, Setsuna does a left circular shift (i.e., the first element in the group will be the last one, and the second element in the group will be the first one, and so on...).

For example, we have  $A = \{1, 2, 3, 4, 5, 6\}$  now. If Setsuna 3-Shift it, it will become  $\{2, 3, 1, 5, 6, 4\}$ .



Now, you have an array  $A$  of length  $n$  (1-indexed). Setsuna will perform two kinds of operations on it:

1. Choose an interval  $[l, r]$  and an integer  $K$ , and  $K$ -Shift the subinterval  $\{a_l, a_{l+1}, \dots, a_r\}$ . It is guaranteed that  $(r - l + 1)$  is a multiple of  $K$ .
2. Choose an interval  $[l, r]$  and ask the value of  $\sum_{i=l}^r a_i$ .

### Input

The first line contains two integers  $n$  and  $m$  ( $3 \leq n, m \leq 2 \times 10^5$ ), which indicates the length of the array  $A$ , and the number of the operations Setsuna will perform.

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

The  $i$ -th of the next  $m$  lines contains four integers  $1, l_i, r_i, K_i$  ( $1 \leq l_i < r_i \leq n, 2 \leq K_i \leq 3, K_i | (r_i - l_i + 1)$ ) or three integers  $2, l_i, r_i$  ( $1 \leq l_i \leq r_i \leq n$ ), which respectively represent two different operations.

### Output

Output the answer to the corresponding operation 2 in each line.

---

**Samples**

Standard Input	Standard Output
6 4	5
1 2 3 4 5 6	20
1 1 4 2	
2 2 3	
1 1 6 3	
2 2 6	

**Note**

After the first operation,  $\{1, 2, 3, 4, 5, 6\}$  will become  $\{2, 1, 4, 3, 5, 6\}$ .

The sum of elements of indexes in  $[2, 3]$  is  $1 + 4 = 5$ .

After the third operation,  $\{2, 1, 4, 3, 5, 6\}$  will become  $\{1, 4, 2, 5, 6, 3\}$ .

The sum of elements of indexes in  $[2, 6]$  is  $4 + 2 + 5 + 6 + 3 = 20$ .

## Problem L. Lottery Tickets

Input file: Standard Input

Time limit: 1 second

Output file: Standard Output

Memory limit: 256 megabytes

Setsuna is almost out of money! Therefore, she always thinks about ways to get rich overnight.

She finds that buying lottery tickets is a wonderful choice, and she decides to use her lucky number as the lottery number.

Setsuna uses some unique methods to pick her lucky numbers. She will first select  $n$  magic cards. Each card is written with a digit between 0 and 9 inclusively. Then she will use some cards to construct a number. She believes that the largest number she can construct which can be divided by 4 is her lucky number.

It is unnecessary to use all the cards, and it is possible that such lucky number does not exist.

Because she is daydreaming of getting rich, only you can help her find this lucky number.

### Input

The input consists of multiple test cases.

The first line contains a single integer  $t(1 \leq t \leq 3 \times 10^5)$ , indicating the number of test cases.

Each of the next  $t$  lines contains 10 integers  $c_0, c_1, \dots, c_9(0 \leq c_i \leq 10^5, \sum_{i=0}^9 c_i > 0)$ , where  $c_i$  indicates the number of cards written with number  $i$ .

It is guaranteed that the sum of  $\sum_{i=0}^9 c_i$  of all test cases does not exceed  $3 \times 10^5$ .

### Output

For each test case given in the input, print the lucky number(without leading zero) or  $-1$  if such lucky number does not exist.

### Samples

Standard Input	Standard Output
6	12
0 1 1 0 0 0 0 0 0 0	4
0 0 0 0 1 0 0 0 0 0	0
1 0 0 2 0 0 0 0 0 0	98765432100
2 1 1 1 1 1 1 1 1 1	9876543120
1 1 1 1 1 1 1 1 1 1	987654312
0 1 1 1 1 1 1 1 1 1	

## Problem M. MITE

Input file: Standard Input

Time limit: 8 seconds

Output file: Standard Output

Memory limit: 256 megabytes

Setsuna is playing a computer game called MITE (MogicianCraft Is Too Easy).

In this game, sugar cane is a valuable plant for crafting rockets and trading paper.

Setsuna has a farm. To be a Farm Tycoon, she wants to plant some sugar cane on the land.

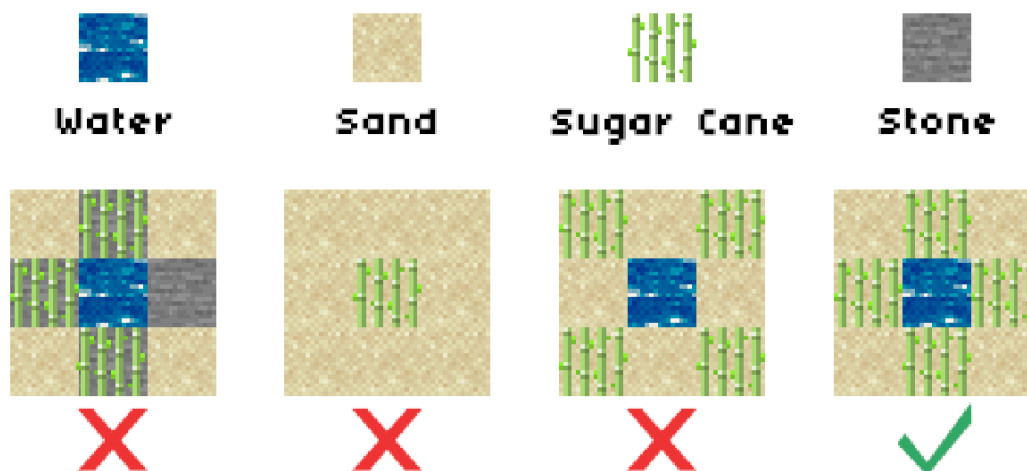
The farm can be seen as a two-dimensional plane composed of  $n \times m$  blocks, and the size of each block is exactly  $1 \times 1$ .

There are four types of blocks: sand block, sand block with sugar cane, rock block and water block.

At the beginning, there are only sand blocks and rock blocks. But, Setsuna can replace sand blocks with water blocks, and she can do this any number of times.

To plant sugar cane on a block, Setsuna should follow these rules:

1. Sugar cane can **only** be planted on sand blocks **on** the farm.
2. The sand block sugar cane planted on must be adjacent to **at least one water block**. Two blocks are considered adjacent if and only if they share an edge.



Setsuna wants to know how to maximize the number of blocks that sugar cane can be planted on.

### Input

The first line of the input contains two integers  $n, m$  ( $1 \leq n \leq 30, 1 \leq m \leq 8$ ).

The next  $n$  lines contains  $m$  characters each, where the  $j$ -th character of the  $i$ -th line is  $b_{i,j}$  ( $b_{i,j}$  is either '.' if the block  $(i, j)$  is a sand block or '#' if the block  $(i, j)$  is a rock block).

### Output

Print  $n$  lines, each of which contains  $m$  characters, where the  $j$ -th character of the  $i$ -th line is  $c_{i,j}$ .

$c_{i,j}$  indicates the type of the block  $(i, j)$  after reforming.

If the block  $(i, j)$  is a water block,  $c_{i,j}$  should be 'O'.

If the block  $(i, j)$  is a sand block with sugar cane,  $c_{i,j}$  should be 'X'.

If the block  $(i, j)$  is just a sand block,  $c_{i,j}$  should be '.'.

If the block  $(i, j)$  is a rock block,  $c_{i,j}$  should be '#'.

You should make sure the number of 'X' is maximal. If there are multiple solutions, print any.

## Samples

Standard Input	Standard Output
3 3 ... .#. ...	XOX X#X OXO
3 3 ... #.# ...	XOX #X# XOX

## Note

Solutions to sample 1 and sample 2 are shown below:

