NSSPC Final 2024

National Taiwan University

2024/09/16

Language	Version	Compile Flags	Extensions
C	gcc 11.3.0	`-O2 -x c -static -lm`	`.c`
C++	g++ 11.3.0	`-O2 -std=gnu++20 -static`	`.cc, .cpp, .cxx, .c++`
Java	openjdk 17.0.12	`[Omitted]`	`.java`

Problem	Problem Name	Points	Time Limit	Memory Limit
A	Nice NSSPC Final	30	1 s	512 MB
B	Work Time	5	1 s	512 MB
C	Baseball League	10	1 s	512 MB
D	String In Strings	15	1 s	512 MB
E	Grading	15	1 s	512 MB
F	Testers	15	1 s	512 MB
G	Star Battle Puzzle	15	1 s	512 MB
H	Repairing Castle	15	1 s	512 MB
I	Table Operations	20	1 s	512 MB
J	Growing Trees	20	1 s	512 MB
K	Rubik's Cube	30	1 s	512 MB
L	Torch Maker	30	1 s	512 MB
M	Experimental Data	30	1 s	512 MB
N	Tidy Up the Desktop	40	1 s	512 MB
O	Ai-chan's Matchsticks	40	1 s	512 MB

This page is intentionally left blank.

A. Nice NSSPC Final

Problem ID: nice

Congratulations on advancing to the NSSPC 2024 final! The contest has started, so let's warm up by outputting a line of "Nice, NSSPC 2024 Final!".

Input

There is no input for this problem.

Output

Output a single line: `Nice, NSSPC 2024 Final!`

Sample Input 1	Sample Output 1
	`Nice, NSSPC 2024 Final!`

This page is intentionally left blank.

B. Work Time

Problem ID: work

WiwiHo cares a lot about time management. She likes to continuously work for X hours and then rest for Y hours. After her rest, she immediately resumes working. For example, if $X = 5$ and $Y = 2$, then WiwiHo would work uninterruptedly for 5 hours then rest for 2 hours.

How many hours in total would WiwiHo work in a span of Z hours, assuming she starts working at the beginning?

Input

The input only contains one line with three integers X, Y, and Z.

- $1 \le X, Y \le 100$
- $1 \le Z \le 10^9$

Output

Output an integer representing the number of hours WiwiHo would be working.

Sample Input 1	Sample Output 1
5 2 20	15

Sample Input 2	Sample Output 2
99 1 24	24

This page is intentionally left blank.

C. Baseball League

Problem ID: league

In the country of Sweet Potato, a baseball league is held every year from March to October. There are six teams in the league, represented by the uppercase letters A, B, C, D, E, F. The league proceeds through a sequence of N events, and there are two types of events:

- Match event: Two different teams play a baseball game, and one of them wins.
- Scoring event: The team with the highest win rate at that moment will receive 1 point. If multiple teams have the same win rate, then **no team will receive a point**. The win rate is calculated as the number of wins divided by the number of matches played by that team. It is guaranteed that at least one game has been played by each of the six teams when this event occurs.

Given all the events in chronological order, calculate the points each team has earned.

Input

The first line of input contains a positive integer N, representing the number of events.

The next N lines describe the events in the order they occurred.

If the i-th event is a match event, the i-th line will contain three strings t_i, x_i, y_i where $t_i =$ `match`, and x_i, y_i are the teams, with x_i being the winning team.

If the i-th event is a scoring event, the i-th line will contain a single string t_i where $t_i =$ `point`.

- $1 \leq N \leq 1000$
- $t_i \in \{\texttt{match}, \texttt{point}\}$
- $x_i, y_i \in \{\text{A}, \text{B}, \text{C}, \text{D}, \text{E}, \text{F}\}$
- $x_i \neq y_i$

Output

Output a single line with 6 integers, representing the points for A, B, C, D, E, F in order.

Sample Input 1

```
11
match A D
match B E
match C F
point
match A B
match B C
point
match C A
point
match B E
point
```

Sample Output 1

```
1 1 0 0 0 0
```

Sample Input 2

```
13
match A C
match B D
match A E
match B F
point
match A D
match B C
point
match A B
match B A
point
match A D
point
```

Sample Output 2

```
1 0 0 0 0 0
```

D. String In Strings

Problem ID: strings

Sam owns two strings s and t that consist of lowercase English letters only. He wants to know if the two strings are similar. In his definition, two strings are similar if they have a **non-empty** common substring.

As the strings can be quite lengthy, Sam asks for your help in finding any **non-empty** common substring of s and t.

A string b is a substring of a if b appears contiguously in a. A string c is a non-empty common substring of a and b if c is not an empty string and c is both a substring of a and a substring of b.

Input

The input consists of two lines. The first line consists of the string s and the second line consists of the string t.

- $1 \leq |s| \leq 10^6$, where $|s|$ denotes the length of s.
- $1 \leq |t| \leq 10^6$, where $|t|$ denotes the length of t.
- All characters are lowercase English letters (`a` to `z`.)

Output

If the strings s and t do not have any non-empty common substring, output "`NO`" (without quotes) in one line.

Otherwise, output two lines. In the first line, output "`YES`" (without quotes). In the second line, output any non-empty common substring of s and t. If there are multiple non-empty common substrings, output any of them.

Sample Input 1	Sample Output 1
looks good	YES oo

Sample Input 2

```
hello
nsspc
```

Sample Output 2

```
NO
```

Sample Input 3

```
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
```

Sample Output 3

```
YES
abcdefghijklmnopqrstuvwxyz
```

E. Grading

Problem ID: grading

Professor Li offers a course called "Advanced Interval Data Structures". Given that the course title includes the word "Advanced", it is not a course that students can simply enroll in; they must pass Professor Li's test to become eligible.

To select students capable of taking this course, Professor Li has devised a task. The correct answer to this task is composed of N strings: $s_1, s_2, \ldots, s_N$. Each string s_i consists of uppercase and lowercase English letters and has a length between 1 and 20. A student's attempt will be considered correct if they send to Professor Li these N strings in **any order**, otherwise, that answer will be considered incorrect.

Students only know that the correct answer to this problem will be 1 to 20 strings composed of uppercase and lowercase English letters, each string having a length between 1 and 20. Formally, a student's attempt will include an integer M, representing the number of strings they believe in the answer, and M strings $t_1, t_2, \ldots, t_M$, representing the strings in their attempt. Their attempt will be considered correct if $N = M$ and there exists a permutation $p_1, p_2, \ldots, p_N$ of 1 to N such that $\forall 1 \le i \le N,\ s_i = t_{p_i}$. In other words, this means that $t_1, t_2, \ldots, t_M$ can be rearranged to become $s_1, s_2, \ldots, s_N$. A permutation of 1 to N is defined as a sequence where each of $1, 2, \ldots, N$ appears exactly once.

Despite the course being insanely challenging, many students still wish to enroll. There are T students who have made their attempts to this problem, and you need to tell Professor Li whether each student's answer is correct.

Input

The first line contains an integer N, representing how many strings are in the correct answer.

The second line contains N strings $s_1, s_2, \ldots, s_N$, representing the N strings in the correct answer, separated by spaces.

The third line contains an integer T, representing the number of students who have submitted their answers.

The next T lines, where the i-th line represents the i-th student's submission, each contain an integer M, representing the number of strings the student believes are in the answer, followed by M strings $t_1, t_2, \ldots, t_M$, separated by spaces.

- $1 \le N, M \le 20$
- $1 \le T \le 10000$
- Each s_i, t_i is a string of uppercase and lowercase English letters with a length between 1 and 20.

Output

Output T lines, where the i-th line contains `Yes` if the i-th student's answer is correct, otherwise output `No`.

Sample Input 1

```
5
good luck and have fun
4
3 luck and fun
5 have fun and good luck
6 good luck and and have fun
5 good luck and have fun
```

Sample Output 1

```
No
Yes
No
Yes
```

Sample Input 2

```
3
Welcome to NSSPC
5
3 WELCOME TO NSSPC
3 to Welcome NSSPC
2 Welcome NSSPC
1 Welcome
3 welcome to nsspc
```

Sample Output 2

```
No
Yes
No
No
No
```

Sample Input 3

```
6
to be or not to be
3
4 to be or not
6 to to be be or not
9 to be or not to be or to be
```

Sample Output 3

```
No
Yes
No
```

F. Testers

Problem ID: testers

NSSPC 2024 is approaching, and the judges must prepare the problems as soon as possible. There are N judges in the judging team, and they plan to prepare a total of N problems, numbered from 1 to N. The author of problem i is judge i.

To ensure the quality of the problems, each problem must be tested by a judge other than the author to confirm that there are no issues. Specifically, each judge must be responsible for testing exactly one problem, and each problem must have exactly one judge responsible for testing it. The judge responsible for testing cannot be the author of the problem. Some judges have already chosen which problems they will test. Since the deadline is approaching, you must quickly assign the remaining judges to the problems they will be responsible for testing.

Input

The first line contains an integer T, representing the number of test cases.

For each test case, the first line contains an integer N, representing the number of judges, and also the number of problems to be prepared.

The second line contains N integers $a_1, a_2, \ldots, a_N$. For $1 \le i \le N$, if $a_i \neq -1$, it means judge i has already chosen to be responsible for testing problem a_i; if $a_i = -1$, it means judge i has not yet chosen which problem to test.

- $1 \le T$
- $2 \le N \le 10^5$
- The total sum of N in all T test cases $\le 10^5$
- $1 \le a_i \le N$ or $a_i = -1$
- It is guaranteed that it is possible to assign problems to all judges with $a_i = -1$ such that all requirements are met

Output

For each test case, output a line containing N integers $b_1, b_2, \ldots, b_N$, where $1 \le b_i \le N$, representing the problem that judge i will be responsible for testing after the assignment. If a judge has already chosen a problem to test, you must keep their original choice. Formally, your output must satisfy all the following conditions:

- $1 \le b_i \le N$
- $b_i \ne i$
- If $a_i \ne -1$, then $b_i = a_i$
- For $1 \le j \le N$, there is exactly one i such that $b_i = j$

Sample Input 1	Sample Output 1
5 5 -1 -1 -1 -1 -1 5 3 4 2 5 1 3 3 -1 -1 3 -1 -1 1 10 3 -1 -1 5 2 -1 1 6 -1 4	2 3 4 5 1 3 4 2 5 1 3 1 2 2 3 1 3 8 10 5 2 9 1 6 7 4

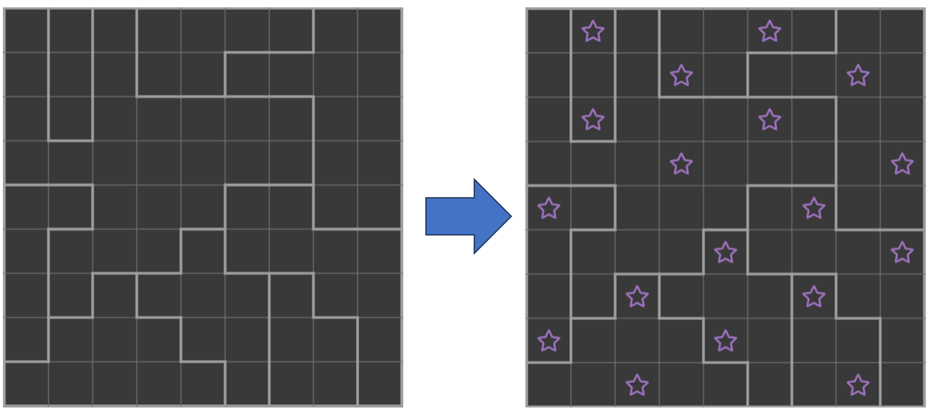
G. Star Battle Puzzle

Problem ID: star-battle

Have you ever played Star Battle Puzzle? Star Battle Puzzle is a logic puzzle game, where each puzzle starts with an initial board: an empty 9×9 grid divided into several regions. The player's goal is to place "stars" into some of the cells, satisfying the following conditions:

- Each row must contain exactly two stars.
- Each column must contain exactly two stars.
- Each region must contain exactly two stars.
- No two stars can be adjacent in any of the eight directions. That is, the cells containing the stars do not share any edge or any corner.

The figure below shows examples of an initial board and the solution of it:



Now, please write a program that, given a Star Battle Puzzle grid with stars already placed, outputs whether the grid is correctly solved. That is, whether the placed stars satisfy all specified conditions.

Input

The input starts with the information of an initial board, consisting of nine lines, each with nine characters, where each character is a digit from `1` to `9`, representing the region number of that cell.

This is followed by the placement of stars, consisting of nine lines, each with nine characters, where each character is either `.` or `*`, where `.` means the cell is empty and `*` means a star is placed in that cell.

- The given empty grid is guaranteed to be composed of characters `1` to `9`.
- The given empty grid is guaranteed to have exactly nine regions, each being four-way connected. That is, you can always move from one cell to another cell from the same region via edge-sharing neighboring cells, with all of them belonging to the same region.
- The given star placement is guaranteed to be composed of characters `.` and `*`.

Output

If the star placement satisfies all conditions, output `Valid`; otherwise, if any condition is violated, output `Invalid`.

Sample Input 1

```
121333344
121334444
121111144
111111144
551116644
511176666
518777966
588877996
888887996
.*...*...
...*...*.
.*...*...
...*....*
*.....*..
....*...*
..*...*..
*...*....
..*....*.
```

Sample Output 1

```
Valid
```

Sample Input 2

```
121333344
121334444
121111144
111111144
551116644
511176666
518777966
588877996
888887996
.*...*...
...*...*.
.*......*
...*.*...
*.....*..
....*...*
..*...*..
*...*....
..*....*.
```

Sample Output 2

```
Invalid
```

Sample Input 3

```
121333344
121334444
121111144
111111144
551116644
511176666
518777966
588877996
888887996
.*.......
...*...*.
.*...*...
...*....*
*.....*..
....*...*
..*...*..
*...*....
..*....*.
```

Sample Output 3

```
Invalid
```

This page is intentionally left blank.

H. Repairing Castle

Problem ID: repair

Little Po has a castle where he lives happily every day until one day Little Yee attacks his castle. During the attack, one wall of the castle is completely destroyed! Coincidentally, Little Po thought that this wall was very ugly, so this gives him a perfect opportunity to rebuild it.

The width of this wall is N meters, and he divides the wall into N sections, each with a width of 1 meter. The sections are numbered from 1 to N from left to right. He orders Q bricks, where each brick has a width of 1 meter, the same thickness as the wall, and varying height. Each brick must fit exactly into one section and cannot span multiple sections or be rotated.

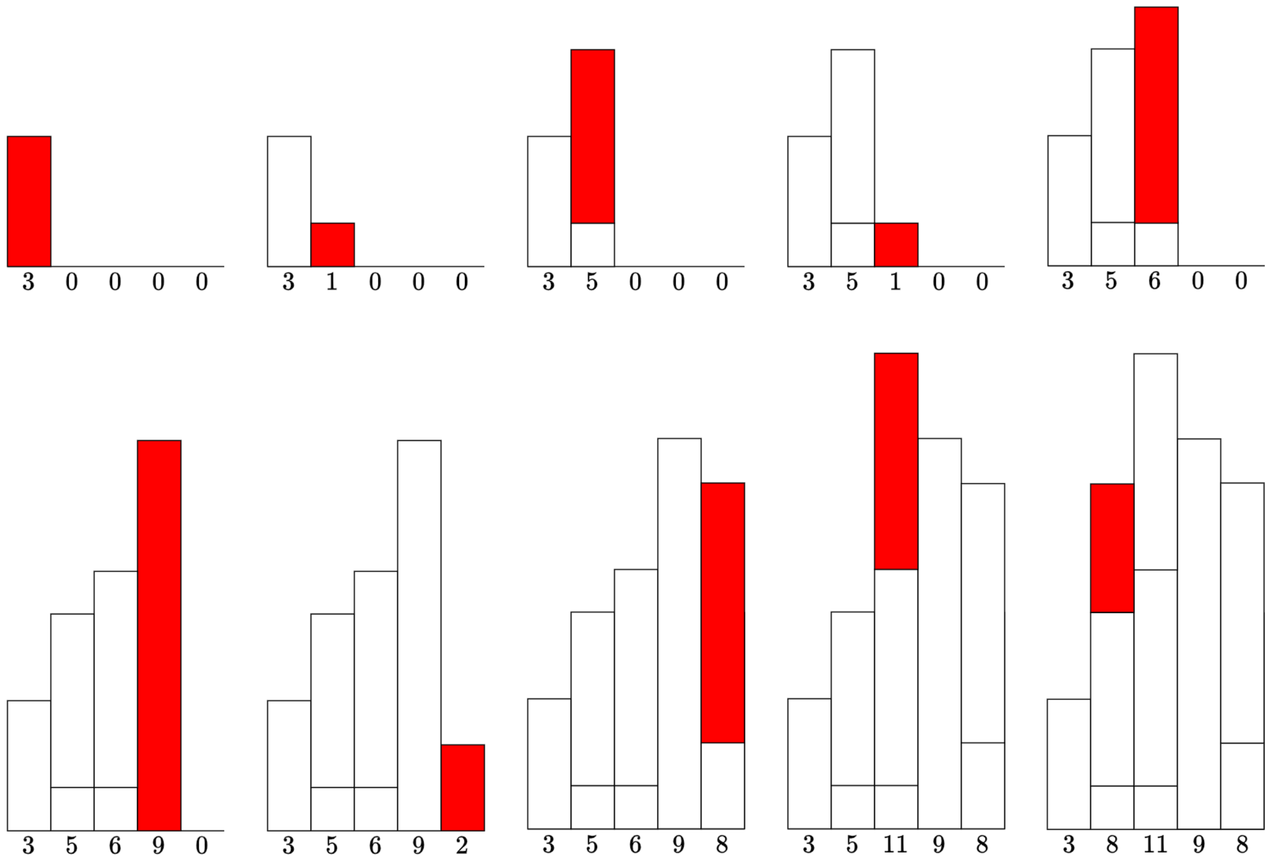
Formally, each of these N sections has its own height. Initially, all sections have heights 0. If a section's height is x originally and Little Po adds a brick with height y to this section, the new height of this section will be $x + y$.

The Q bricks will arrive one by one. The height of the i-th brick is h_i, but Little Po doesn't know the height of each brick until it arrives. He's eager to see the wall completed, so he must place each brick immediately upon arrival. The principle by which he chooses a section is as follows: If the current height of the i-th section is x_i, the **ugliness** of the wall is defined as

$$\sum_{k=0}^{N}(x_k - x_{k+1})^2$$

where $x_0 = x_{N+1} = 0$. Little Po wants to choose a section to place the brick so that the resulting ugliness is minimized. If multiple sections result in the same minimum ugliness, he will choose the smallest numbered section.

For example, if $N = 5$, $Q = 10$, and the heights of the 10 bricks are $3, 1, 4, 1, 5, 9, 2, 6, 5, 3$ in order. When placing the first brick, no matter where it is placed, the ugliness will be 18, so Little Po will place the first brick in the first section. When placing the second brick, if placed in section 1, the ugliness will be 32; if placed in section 2, the ugliness will be 14; and if placed in any other section, the ugliness will be 20. Therefore, Little Po will place the second brick in section 2. The changes in the height of each section during the process are shown in the figure below, with the red bricks representing the newly placed bricks, and the numbers below representing the height of each section.



Little Yee secretly knows the heights of these Q bricks and is curious about what Little Po's newly built wall will look like so he can plan his next attack. Please tell Little Yee the final height of each section.

Input

The first line contains two integers N and Q, representing the width of the wall and the number of bricks, respectively.

The second line contains Q integers $h_1, h_2, \dots, h_Q$, representing the heights of the Q bricks in the order they arrive.

- $1 \le N, Q \le 500$

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

Output

Output N integers $x_1, x_2, \ldots, x_N$, representing the final height of the i-th section is x_i.

Sample Input 1	Sample Output 1
5 10 3 1 4 1 5 9 2 6 5 3	3 8 11 9 8

Sample Input 2	Sample Output 2
3 4 100000 100000 100000 100000	100000 200000 100000

This page is intentionally left blank.

I. Table Operations

Problem ID: operations

You are given an $N \times M$ table, with each cell containing a lowercase letter. We use (i, j) to represent the cell in the i-th row $(1 \le i \le N)$ from the top and j-th column $(1 \le j \le M)$ from the left. You need to perform Q operations on the table in sequence. There are two main types of operations.

The first type of operation is "rotation". Each rotation operation is accompanied by a positive integer t, which indicates that the entire table should be rotated clockwise by 90 degrees t times consecutively. After rotating a $R \times C$ table clockwise by 90 degrees, it will become a $C \times R$ table, and the cell originally at (i, j) will move to $(j, R+1-i)$. For example, the 2×3 table on the left will become the 3×2 table on the right after rotating 90 degrees once.

a	b	c
d	e	f

$\longrightarrow$

d	a
e	b
f	c

The second type of operation is "flip", which can be further divided into four types of flip operations: horizontal flip, vertical flip, and flips along the main and anti-diagonals. The effects of these four types of flips are as follows:

- Horizontal flip: After a horizontal flip, an $R \times C$ table remains an $R \times C$ table, but the cell originally at (i, j) moves to $(i, C+1-j)$.
- Vertical flip: After a vertical flip, an $R \times C$ table remains an $R \times C$ table, but the cell originally at (i, j) moves to $(R+1-i, j)$.
- Main diagonal flip: After a main diagonal flip, an $R \times C$ table becomes a $C \times R$ table, and the cell originally at (i, j) moves to (j, i).
- Anti-diagonal flip: After an anti-diagonal flip, an $R \times C$ table becomes a $C \times R$ table, and the cell originally at (i, j) moves to $(C+1-j, R+1-i)$.

For example, the 2×3 table on the left will become the four tables on the right after horizontal, vertical, main diagonal, and anti-diagonal flips, respectively.

a	b	c
d	e	f

$\longrightarrow$

c	b	a
f	e	d

d	e	f
a	b	c

a	d
b	e
c	f

f	c
e	b
d	a

After performing all the operations, please provide the final appearance of the table.

Input

The first line contains three integers N, M, and Q.

Next, there are N lines, where the i-th line contains a string s_i of length M. The j-th character in s_i represents the lowercase letter in cell (i, j).

Next, there are Q lines, where the i-th line contains a character op_i and an integer t_i with the following meanings:

- If $op_i = \mathsf{R}$, then the i-th operation is a rotation operation. The table should be rotated 90 degrees t_i times consecutively.
- If $op_i = \mathsf{F}$, then the i-th operation is a flip operation, where $t_i = 1, 2, 3, 4$ corresponds to a horizontal flip, a vertical flip, a main diagonal flip, and an anti-diagonal flip, respectively.

- $1 \leq N, M, Q \leq 100$
- All s_i consist of only lowercase letters
- op_i is either R or F
- If $op_i = \mathsf{R}$, then $1 \leq t_i \leq 10^9$.
- If $op_i = \mathsf{F}$, then $1 \leq t_i \leq 4$.

Output

On the first line, output two positive integers N', M' indicating that after all operations, the resulting table has size $N' \times M'$.

Next, output N' lines, where the i-th line contains a string s'_i of length M'. The j-th character in s'_i represents the lowercase letters in cell (i, j) of the final table.

Sample Input 1

```
3 10 4
baluteshih
littlecube
hsinmutsai
F 3
R 2
F 1
R 103
```

Sample Output 1

```
3 10
iastumnish
ebucelttil
hihsetulab
```

Sample Input 2

```
3 6 4
wiwiho
howiwi
wihowi
F 4
F 2
R 49
R 1
```

Sample Output 2

```
6 3
oii
hww
iio
wwh
ioi
whw
```

This page is intentionally left blank.

J. Growing Trees

Problem ID: tree

Old MacDonald has a piece of farm that can be represented as an $N \times M$ grid. Initially, there is a tree of height 0 planted at each position in the grid. Let's number the rows from top to bottom and the columns from left to right, and denote by (i, j) the cell in the i-th row and j-th column. Old MacDonald has many growth agents, and he can choose a cell (i, j) that satisfies $2 \le i \le N - 1$ and $2 \le j \le M - 1$ to use the growth agent. After using it, the height of the trees in all cells (x, y) that satisfy $|x - i| \le 1$ and $|y - j| \le 1$ will increase by 1.

This morning, Old MacDonald used several growth agents. Unfortunately, by the afternoon, he forgot where he had used them. Now Old MacDonald tells you the height of each tree, and he wants you to help him determine the number of times he used the growth agent at each position.

Input

The first line of the input contains two positive integers N and M, representing the size of the grid.

The next N lines contain M numbers each. The i-th line contains $a_{i,1}, a_{i,2}, \ldots, a_{i,M}$, where $a_{i,j}$ represents the height of the tree in the cell (i, j).

- $3 \le N \le 500$
- $3 \le M \le 500$
- $0 \le a_{i,j} \le 10^6$
- It is guaranteed that Old MacDonald didn't miscount the height of the trees, and the growth agent is the only way to increase the height of the trees.

Output

Output $N - 2$ lines. The i-th line contains $M - 2$ integers $b_{i,1}, b_{i,2}, \ldots, b_{i,M-2}$, where $b_{i,j}$ represents the number of times Old MacDonald used the growth agent in the cell $(i + 1, j + 1)$.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

```
4 3
0 0 0
0 0 0
0 0 0
0 0 0
```

Sample Output 2

```
0
0
```

K. Rubik's Cube

Problem ID: rubiks-cube

FHVirus has always been fascinated by Rubik's cubes. To be proficient in solving various types of Rubik's cubes, one must not only be able to solve them but also solve them quickly. Because of this, FHVirus not only explored different kinds of Rubik's cubes but also participated in various Rubik's Cube competitions.

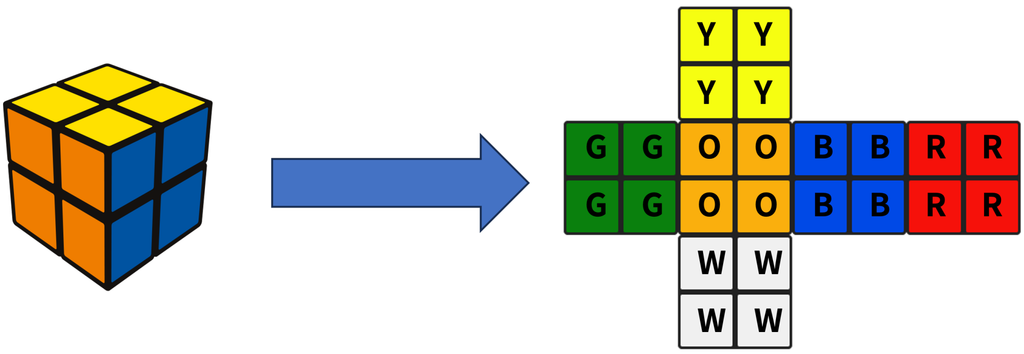
As Rubik's cubes lover, FHVirus recently started learning programming. Combining his interest in Rubik's cubes with his newfound passion for programming, he decided to write a "Rubik's Cube scrambler" to help him practice.

What is a Rubik's Cube scrambler? Rubik's Cube enthusiasts need to repeatedly scramble and quickly solve the Cube to practice. If the Cube is scrambled arbitrarily each time, human habits inevitably lead to similar scramble patterns. Therefore, writing a fair "scrambler" is very important.

Thus, a "Rubik's Cube scrambler" generates a random sequence of "scrambling algorithms," which players can follow to scramble the cube. After scrambling, players then practice solving it quickly in their usual manner.

However, FHVirus just started learning programming, so he decided to begin with the simplest "2x2 Rubik's Cube scrambler."

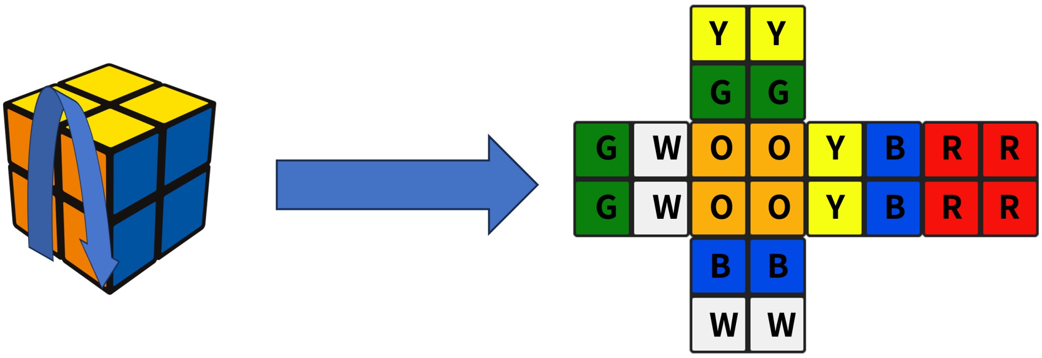
First, we need to understand what a "2x2 Rubik's cube" is, as shown in the image below:



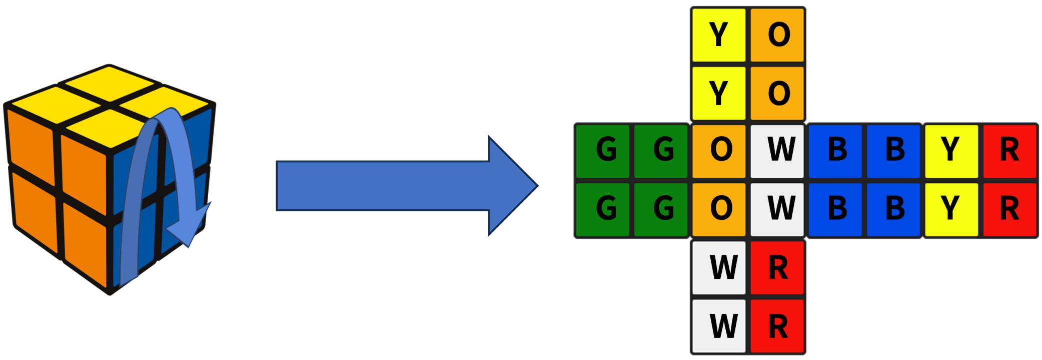
In the image above, the left side shows a 2x2 Rubik's cube, which has six faces, each consisting of four squares, commonly colored yellow, orange, blue, red, green, and white. The right side is its unfolded diagram, showing the colors of all six faces, with yellow, orange, blue, red, green, and white abbreviated as `Y`, `O`, `B`, `R`, `G`, and `W`, respectively.

Next, we introduce the three basic operations on the 2x2 Rubik's cube.

1. F operation
   The F operation rotates the front face of the Rubik's Cube clockwise, as shown below:



   In the image above, the left side shows the rotating face and direction, and the right side shows the unfolded diagram after the rotation.
   To simplify notation, F2 indicates "rotating the front face of the Rubik's Cube 180 degrees," which is equivalent to performing the F operation twice; F' indicates "rotating the front face of the Rubik's Cube counterclockwise," which is equivalent to performing the F operation three times.
2. R operation
   The R operation rotates the right face of the Rubik's Cube clockwise, as shown below:

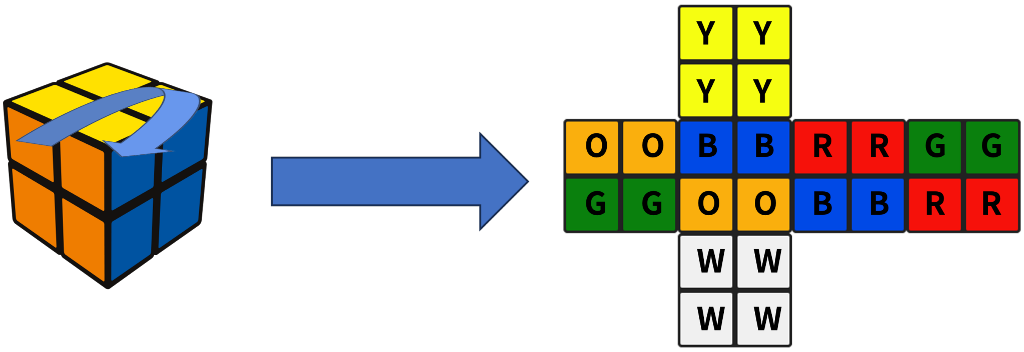


   In the image above, the left side shows the rotating face and direction, and the right side shows the unfolded diagram after the rotation.
   To simplify notation, R2 indicates "rotating the right face of the Rubik's Cube 180 degrees," which is equivalent to performing the R operation twice; R' indicates "rotating the right face

of the Rubik's Cube counterclockwise," which is equivalent to performing the R operation three times.

3. U operation

The U operation rotates the top face of the Rubik's Cube clockwise, as shown below:



In the image above, the left side shows the rotating face and direction, and the right side shows the unfolded diagram after the rotation.

To simplify notation, U2 indicates "rotating the top face of the Rubik's Cube 180 degrees," which is equivalent to performing the U operation twice; U' indicates "rotating the top face of the Rubik's Cube counterclockwise," which is equivalent to performing the U operation three times.

With these basic operations at hand, it's time to write the 2x2 Rubik's Cube scrambler. First, we need to write a "scramble algorithm generator" to generate a sequence of basic operations. Then, we need to write a "rotation simulator" to simulate the state of the 2x2 Rubik's Cube after following the scrambling algorithm, so players can verify if they scrambled correctly.

FHVirus has already written the generator part, but he encountered difficulties with the simulator part. Can you help him write a simulator that reads the scrambling algorithm and outputs the state of the 2x2 Rubik's Cube after following the scrambling algorithm?

Input

The first line of input contains a positive integer N, representing the length of the scrambling algorithm generated.

The next line contains N strings, each representing one of the nine basic operations: `F`, `F'`, `F2`, `R`, `R'`, `R2`, `U`, `U'` and `U2`.

- $1 \leq N \leq 100$

Output

Output the unfolded diagram of the Rubik's Cube after following the scrambling algorithm, starting from the initial state shown in the first image in the problem statement.

For the unfolded diagram, the judging system will perform a strict comparison, meaning you must output according to the following format:

```
  YY
  YY
GGOOBBRR
GGOOBBRR
  WW
  WW
```

The first, second, fifth, and sixth lines each have two leading space characters (ASCII code 32), and each line ends with a newline character (ASCII code 10). Note that the first, second, fifth, and sixth lines do not have extra trailing spaces, making their lengths shorter by four characters compared to the third and fourth lines.

Sample Input 1

```
8
U F R F R U F R
```

Sample Output 1

```
  WB
  RG
OBWWORYB
GYGOYGYR
  RB
  WO
```

Sample Input 2

```
10
F2 R' U F' R2 U2 F R' F' R
```

Sample Output 2

```
  OY
  RG
GBWYROBY
GRBBWOWR
  YO
  WG
```

This page is intentionally left blank.

L. Torch Maker

Problem ID: torch

LittleCube loves playing cube games and has recently discovered a new cube game. In this game, there are five types of items: logs, woods, charcoal, sticks, and torches. LittleCube currently has N logs and he loves torches very much, so he wants to make as many torches as possible using these logs. How can LittleCube make torches? He can perform two types of operations: **crafting** and **smelting**.

- Crafting: By consuming the required materials according to the recipe, he can obtain the specified item. Here are all the crafting recipes, where consuming the items on the left side of the arrow will yield the items on the right side.

$$\begin{aligned} \text{1 log} &\rightarrow \text{4 woods} \\ \text{2 woods} &\rightarrow \text{4 sticks} \\ \text{1 stick and 1 charcoal} &\rightarrow \text{1 torch} \end{aligned}$$

- Smelting: One smelting operation can convert 1 log into 1 charcoal, but it requires 1 unit of energy. Energy can be obtained by consuming fuel. All items except torches can be used as fuel. The table below shows how many units of energy he will obtain when consuming each type of item as fuel:

Fuel	Energy
1 log	1 unit
1 wood	1 unit
1 stick	0.2 units
1 charcoal	8 units

  Note that to use an item as fuel, the entire item must be consumed. For example, you cannot consume only 1/8 of a charcoal to obtain 1 unit of energy.

For example, if LittleCube starts with 10 logs, he can first craft 3 logs into $3 \times 4 = 12$ woods. Using 7 of these woods as fuel will give him 7 units of energy, which he can use to smelt the remaining 7 logs into charcoal. He can then craft 4 of the remaining 5 woods into $\frac{4}{2} \times 4 = 8$ sticks. Finally, he can use 7 sticks and 7 charcoals to craft 7 torches.

Please tell LittleCube the maximum number of torches he can make.

Input

The first line contains an integer T, representing the number of test cases.

The next T lines each contain a single integer N, representing the number of logs LittleCube has.

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 10^{12}$

Output

Output T lines, where the i-th line contains a single integer representing the answer for the i-th test case.

Sample Input 1	Sample Output 1
14 1 2 3 4 5 6 7 8 9 10 100 271828182845 314159265359 1000000000000	0 1 2 2 3 4 4 5 6 7 78 214399693511 247787589578 788732394365

M. Experimental Data

Problem ID: experiments

Charlie has been conducting some research recently. This research involves performing several experiments, with each experiment yielding a score representing its result. Charlie conducted a total of N experiments, where the score of the i-th experiment is a_i.

Charlie needs to analyze the data he has obtained. During the analysis, he needs to compute the variance for certain intervals. More specifically, there are Q pairs of intervals (l_i, r_i). For each pair (l_i, r_i), Charlie needs to compute the variance of $a_{l_i}, a_{l_i+1}, \ldots, a_{r_i}$.

Charlie doesn't know how to quickly compute these Q variances, so he has come to you for help. Please assist Charlie in calculating the Q variances he needs.

The variance of K real numbers $x_1, x_2, \ldots, x_K$ is defined as $\frac{1}{K}\sum_{i=1}^{K}(x_i - \bar{x})^2$, where $\bar{x} = \frac{1}{K}\sum_{i=1}^{K} x_i$

Input

The first line contains two integers N and Q.

The second line contains N integers $a_1, a_2, \ldots, a_N$.

The next Q lines each contain two integers l_i and r_i.

- $1 \le N, Q \le 2 \cdot 10^5$
- $-10^6 \le a_i \le 10^6$
- $1 \le l_i \le r_i \le N$

Output

Output Q lines. The i-th line should contain an integer representing $(r_i - l_i + 1)^2 \cdot V_i$, where V_i is the variance of $a_{l_i}, a_{l_i+1}, \ldots, a_{r_i}$.

It can be proven that $(r_i - l_i + 1)^2 \cdot V_i$ is always an integer under the given constraints.

Sample Input 1

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

Sample Output 1

```
24
0
50
```

Sample Input 2

```
6 3
-49 49 49 -49 100 -100
1 6
2 3
4 6
```

Sample Output 2

```
177624
0
64802
```

N. Tidy Up the Desktop

Problem ID: desktop

Have you ever accidentally placed too many files on your computer desktop, making it extremely cluttered? People who are used to keeping their environment tidy might not experience this, but 8e7 is not one of those people. One day, 8e7 realized that his desktop was incredibly messy, largely because he had left too many unnecessary files on it as temporary storage. By deleting these files, his desktop could become significantly cleaner.

However, 8e7's reluctance to keep his environment tidy is ultimately due to laziness. Therefore, he wants to expend the least effort to delete these files. 8e7 discovered that by selecting a rectangle on the desktop, he could turn those files within that rectangle into selection status. Clicking delete would then remove all selected files in one go. But upon selecting a second rectangle, he realized things were not that simple!

In fact, 8e7's computer's operating system is designed such that whenever he selects a rectangle on the desktop, the selection status of all files within that rectangle is toggled.

What does this mean? It means that any file within the rectangle that was previously unselected becomes selected, and any file that was already selected becomes unselected!

This operation becomes quite complex, but 8e7 still wants to use the least effort to select all files so he can delete them with one click. Now, 8e7 has converted all the files into N coordinates on a 2D plane, where M of these files need to be deleted. Please write a program that, after reading the coordinates of all files, provides 8e7 with a method to select all files using the **fewest rectangle selection operations**.

Input

The first line of input contains two positive integers N and M, representing the total number of files on 8e7's desktop and the number of files that need to be deleted.

In the next N lines, the i-th line contains two integers x_i and y_i, representing the coordinates of the i-th file. The files numbered 1 through M are the files that 8e7 wishes to delete.

- $1 \leq M \leq N \leq 14$
- $-10^9 \leq x_i, y_i \leq 10^9$
- It is guaranteed that all file coordinates are distinct.

Output

The first line of output should contain a positive integer k, representing the number of rectangle selection operations needed.

In the next k lines, the j-th line should contain four integers l_j, d_j, r_j, u_j, representing the coordinates of the rectangle formed by (l_j, d_j) and (r_j, u_j) for the j-th selection operation. This rectangle must satisfy $-2 \times 10^9 \leq l_j \leq r_j \leq 2 \times 10^9$ and $-2 \times 10^9 \leq d_j \leq u_j \leq 2 \times 10^9$. For the i-th file, if $l_j \leq x_i \leq r_j$ and $d_j \leq y_i \leq u_j$, its selection status will be toggled.

Your output selection method must ensure that the files numbered 1 through M are in the selected state, while the other files remain unselected. If there are multiple possible solutions, output any one of them as they will all be considered correct.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

```
6 3
-3 5
-3 9
6 0
6 5
-3 8
6 2
```

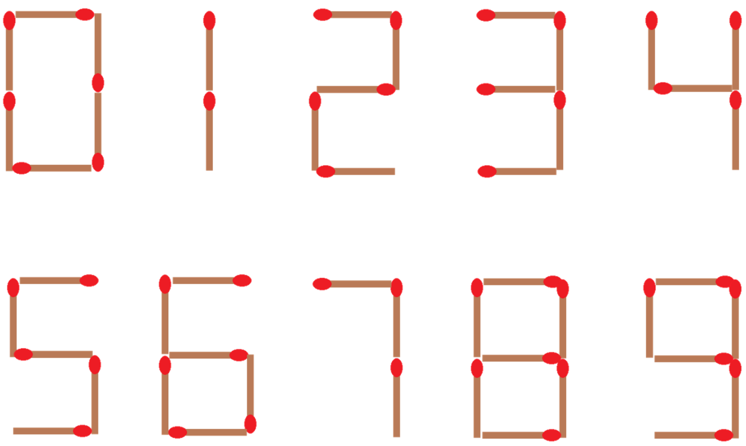
Sample Output 2

```
3
-3 0 6 0
-3 9 -3 9
-3 0 -3 5
```

O. Ai-chan's Matchsticks

Problem ID: matchsticks

It is well known that matchsticks can be used to make numbers. The ways to make the digits from 0 to 9 with matchsticks are shown in the following figure:



Different integers might require different numbers of matchsticks. For example, to make the number 17, one needs a 1 and a 7, requiring a total of $2+3=5$ matchsticks. To make the number 45510, one would need $4+5+5+2+6=22$ matchsticks.

Ai-chan has N matchsticks and wants to use all of them to make a single non-negative integer.

How many different non-negative integers can Ai-chan make? Since the answer can be very large, please output the remainder of the answer divided by 998244353.

Input

The input consists of a single positive integer N.

- $1 \le N \le 10^{18}$

Output

Please output a non-negative integer representing the number of different non-negative integers Ai-chan can make, modulo 998244353.

Sample Input 1	Sample Output 1
7	12

Sample Input 2	Sample Output 2
3	1

Sample Input 3	Sample Output 3
1000000	811496257