Hong Kong Olympiad in Informatics 2013
Junior Group

Task Overview

Task	CPU time limit	Score
Enumeration	1 second	50
Strategy Steps	1 second	100
Queueing	1 second	100
Dryads	1 second	100
Lucky Rainbow	1 second	100

Notice:
C++ programmers should be aware that using C++ streams (cin / cout) may lead to I/O bottlenecks and substantially slower performance.

C/C++ programmers should use `"%I64d"` for 64-bit integers I/O.

Enumeration

Time Limit: 1 second

Problem

Write a program to read an integer N and output N^2 integers in the format specified below.

Input

The input contains an integer N ($1 \leq N \leq 10$).

Output

The output consists of N lines, each containing N integers.
The first integer in the i-th line is the i-th square number. For the other integers, each of them is the next square number of its left.

Sample test

Input	Output

4	1 4 9 16
	4 9 16 25
	9 16 25 36
	16 25 36 49

Strategy Steps

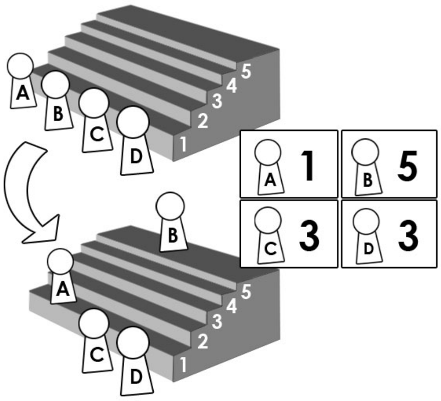
Time Limit: 1 second

Problem

You and your classmates (M people in total) had fun during recess. When the bell rang, you approached the stairs to go back to your classroom. Suddenly, you recalled the ‘Strategy Steps’ mini game from Wii Party, and have decided to ask them to play this with you.

There are N steps from ground floor to your classroom. In a round, each of the players may pick a number: 1, 3 or 5. If his number is not picked by any other player, he/she may move up that number of steps. The procedure is repeated until a player has reached the top of the stairs (his remaining steps, if any, will be discarded). The game ends after that. If two or more players reach the top in the same round, all of them are winners.

For example, in a 4-player game, players A, B, C and D picked 1, 5, 3 and 3 respectively. Player A can then move up 1 step and player B can move up 5 steps.



Unfortunately because it is the end of recess, you do not have enough time to finish the game. Therefore, you and your classmates came up with an easy way to decide the winner: each of you will write down 100 numbers (1, 3 or 5) in sequence, representing the numbers you would pick. Then you will use a program to simulate the game and decide who is/are the winner(s), and the ending positions of each player when the game ends. You may assume that the game will end within the 100 rounds.

Input

The first lines contains two integers N and M.
The following 100 lines each contains M integers (1, 3 or 5). The i-th integer on the j-th line is the number picked by Player i for Round j.

Output

The first line is an integer, the number of rounds completed when the game ends.
The second line consists of M integers, the k-th integer is the number of steps player k moved up.

Sample test

Input	Output
10 4 1 5 3 3 5 1 1 5 3 5 1 3 ··· (97 more lines)	3 1 10 1 0

7 2 1 1 3 3 1 5 3 1 5 5 1 3 ··· (94 more lines)	6 5 7

Constraints

In test cases worth 25% of the total points, $M = 2$.
In test cases worth 25% of the total points, $M = 3$.
In test cases worth 25% of the total points, $4 \leq M \leq 30$.
In all test cases, $1 \leq N \leq 500$, $2 \leq M \leq 10000$.

Queueing

`Time Limit: 1 second`

Problem

After school, you and your classmates decided to visit the newly opened gaming centre called 8-bit Gym H.K. It features a lot of stall games and electronic gaming machines. Due to popularity, there is a long queue formed.

8-bit GYM H.K.

8-bit Gym H.K. has a membership system. Each member will receive an 8-digit identification number. Gold members' numbers start with '1' and ordinary members' start with '0'. Non-members are also given a 12-digit identification number when they join the queue.

People are allowed to enter the centre one by one in a first-come-first-served manner, except under the following situations:

- If the person at the front of the queue is not a member and the second person is a member, and if the first person had been stalled fewer than two times, he/she will be stalled and the second person will be served first.
- If the person at the front of the queue is an ordinary member and the second person is a gold member, and if the first person had never been stalled, he/she will be stalled and the second person will be served first.

There is N people in queue now. Write a program to find out who is served at specified times.

Input

The first line is a single integer N.
The second line consists of N identification numbers from the front to the end of queue.
The third line is a single integer M.
The fourth line consists of M distinct integers $A_1, A_2, \cdots, A_M$.

Output

You should output M identification numbers. The i-th number should identify the A_i-th person served.

Sample test

Input	Output

4
09876543 12345678 123456789012 210987654321
2
1 4
	12345678 210987654321

Input

6
456789012345 12345678 345678901234 09876543 01234567 210987654321
3
5 2 6

Output
345678901234 456789012345 210987654321

Constraints

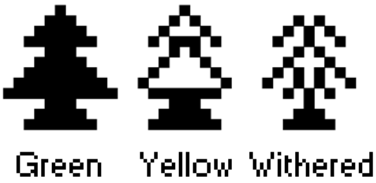
In test cases worth 50% of the total points, $1 \leq M \leq N \leq 1000$.
In all test cases, $1 \leq M \leq N \leq 50000$, $1 \leq A_i \leq N$.

Dryads

Time Limit: 1 second

Problem

There is a video game called 'Dryads' in 8-bit Gym H.K. In Greek mythology, a Dryad is the spirit a tree. The game play is as follow. On the screen, N trees are shown a row. Each of the trees is either Green, Yellow or Withered:



Now you are playing as Dryads to turn all trees into Green using spells. Each time you cast a spell, you can apply on a continuous segment of trees (from the L-th tree to the R-th tree inclusive). The spells available are:

- `Grow(L, R)`: Withered becomes Yellow, Yellow becomes Green. Must not apply on Green.
- `Blow(L, R)`: Yellow becomes Withered. Must not apply on Green or Withered.
- `SuperGrow(L, R)`: Withered becomes Green. Must not apply on Yellow or Green.

The number of tickets you win depends on how well you play the game. Let T be the maximum number of tickets you can win in a single game. Let Opt be the optimal (minimum possible) number of spells required and C be the number of spells you used. The number of tickets t you will get is calculated as follow:

$$t = \begin{cases} T & \text{if } C = Opt \\ (0.1 + 0.4\frac{N-C}{N-Opt})T & \text{if } Opt < C \leq N \\ 0 & \text{if } C > N \end{cases}$$

Note that if you cast a spell illegally, the game is over immediately and you will not get any tickets in that game. The tickets you win in each game will keep as floating point numbers and summed up for all games. Then it will be rounded and you will get that amount of tickets which is also your score of this task.

Input

The first line is a single integer N.
The second line consists of N characters, representing the trees on the screen. `G` means Green, `Y` means Yellow and `W` means Withered.

Output

The sequence of spells you would cast in the format: `<Spell> L R`. Spell names are **case-sensitive**. Your output should always be terminated by a line with the word `Finish`.

Sample test

Input	Output
10 YGYYWWWYGG	Grow 1 1 Grow 3 8 Grow 5 7 Finish

Input	Output
8 WWYYWWYY	Blow 3 4 Blow 7 8 SuperGrow 1 8 Finish

Constraints

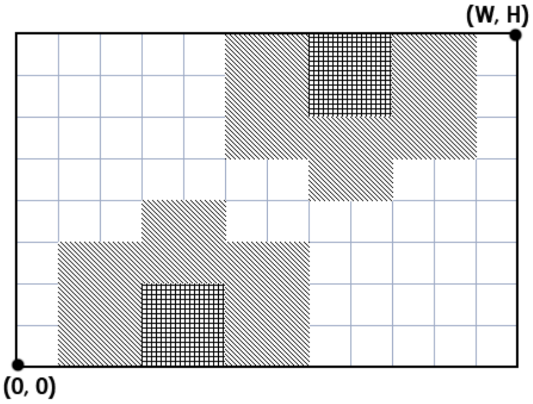
In test cases worth 30% of the total points, $1 \leq N \leq 300$.
In test cases worth 60% of the total points, $1 \leq N \leq 2000$.
In all test cases, $1 \leq N \leq 100000$.

Lucky Rainbow

Time Limit: 1 second

Problem

One of its signature games in 8-bit Gym H.K. is the ‘8-bit Lucky Rainbow’. Unlike traditional lucky rainbows, the rainbows have straight, perpendicular edges. An example of a 12×8 ‘8-bit Lucky Rainbow’ target face is shown below:



The bottom left corner of the target is defined as $(0, 0)$ and the top right corner is defined as (W, H). The background of the target face is painted white, indicated by the character `Z`. Then the target face is painted with exactly P other colors. The i-th color is indicated by the i-th English alphabet and its prize is T_i tickets. Therefore, the sample target face can also be represented with the following:

```
ZZZZZAABBAAZ
ZZZZZAABBAAZ
ZZZZZAAAAAAZ
ZZZZZZZAAZZZ
ZZZAAZZZZZZZ
ZAAAAAAZZZZZ
ZAABBAAZZZZZ
ZAABBAAZZZZZ
```

You have got yourself R square-shaped (of course!) tokens to throw. The side length of the token is L. When it lands, the token’s edges will always parallel or perpendicular to the borders. The bottom-left corner of the i-th token thrown is at (X_i, Y_i).

In order to win tickets, your token must land entirely within a region of a single color (except the white background). To be more specific: You win the prize of color C if no area of the token is out of bounds or overlap with any color other than C.

Now, write a program to find out the total number of tickets you won.

Input

The first line consists of 5 integers: W, H, P, R, L.
The second line consists of P integers, the number of prize tickets for each color, starting from color `A`.
The next H lines each consists of W characters, representing the grid of target.
The next R lines each consists of two numbers X_i and Y_i.

Output

A single integer, the total number of tickets won.

Sample test

Input	Output
12 8 2 5 2 10 15 ZZZZZAABBAAZ ZZZZZAABBAAZ ZZZZZAAAAAAZ ZZZZZZZAAZZZ ZZZAAZZZZZZZ ZAAAAAAZZZZZ ZAABBAAZZZZZ ZAABBAAZZZZZ 1 0 7 6 5 0.23 8.2 5.1 5 7	35

Explanation

$(1, 0) \rightarrow A$
$(7, 6) \rightarrow B$
$(5, 0.23) \rightarrow A$
$(8.2, 5.1) \rightarrow$ Token overlaps A, B and the white background.
$(5, 7) \rightarrow$ Token partially out-of-bounds.

Constraints

In test cases worth 40% of the total points, X_i and Y_i are integers.
In test cases worth 60% of the total points, $1 \leq W, H, R \leq 100$.
In all test cases, $1 \leq W, H \leq 1000$, $1 \leq P \leq 25$, $1 \leq L \leq 100$, $1 \leq R \leq 10000$, $-100 \leq X_i \leq W$, $-100 \leq Y_i \leq H$, $1 \leq T_i \leq 1000$.