BIFRÖST

Heimdall is planning to make his yearly journey along *Bifröst*, the rainbow path to the heavens. The path consists of k cities, and consecutive cities in the path are connected by *bridges*. Each bridge is a certain colour, denoted by a non-negative integer between 0 and 5 000 inclusive. The $k-1$ bridges are numbered 1 to $k-1$ from left to right and the ith bridge is coloured c_i.

Heimdall's journey starts in the leftmost city and ends in rightmost city, possibly revisiting some cities more than once. In total, he crossed bridges A times. The ith bridge he crossed was coloured a_i, and he wrote this down in his notebook.

For example, if there were $k = 8$ cities and the bridges were coloured $1, 3, 2, 3, 4, 4, 5$, then the colours written in his notebook could be:

- $1, 3, 2, 3, 4, 4, 5$.
- $1, 3, 3, 1, 1, 3, 2, 3, 4, 4, 5$.
- $1, 3, 2, 3, 4, 4, 4, 4, 3, 3, 4, 4, 5, 5, 5$.

These are shown in the figure below.

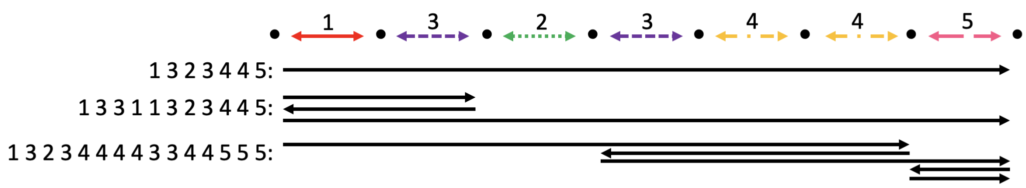

Figure 1: Three possible journeys. The colour of each bridge is written above it.

This year, Loki also made a journey from the leftmost city to the rightmost city. In total, he crossed bridges B times, the ith one coloured b_i. He wrote this down in his notebook too.

Heimdall thinks Loki might be lying, so he has asked you to determine if there is any integer k and sequence $c_1, c_2, \ldots, c_{k-1}$ that agrees with both his and Loki's notebooks.

Subtasks and Constraints

For all subtasks:

- $1 \leq A, B \leq 5\,000$.
- $0 \leq a_i, b_i \leq 5\,000$ for all i.

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	6	$a_i = 0$ and $b_i = 0$ for all i.
2	10	Each colour appears at most once in Heimdall's notebook.
3	16	Each colour (except 0) appears at most once in Heimdall's notebook.
4	17	$A, B \leq 15$
5	34	$A, B \leq 100$
6	17	$A, B \leq 5000$

Input

- The first line of input contains the two integers A and B.
- The second line contains A integers $a_1, a_2, \ldots, a_A$.
- The third line contains B integers $b_1, b_2, \ldots, b_B$.

Output

If there is no sequence that agrees with both Heimdall and Loki's notebooks, output the single line `IMPOSSIBLE`.

Otherwise, output two lines:

- On the first line, print k, the number of cities. Your choice of k must be at most 5001.
- On the second line, print the $k-1$ integers $c_1, c_2, \ldots, c_{k-1}$.

If there are multiple correct sequences, you can print any of them. You **do not** need to minimize nor maximize k.

Sample Input 1

```
11 15
1 3 3 1 1 3 2 3 4 4 5
1 3 2 3 4 4 4 4 3 3 4 4 5 5 5
```

Sample Output 1

```
8
1 3 2 3 4 4 5
```

Sample Input 2

```
4 3
5 0 0 3
5 0 3
```

Sample Output 2

```
IMPOSSIBLE
```

Explanation

Sample Input 1 corresponds to the case shown in Figure 1. One possible solution has $k = 8$ cities, with the $k - 1$ bridges coloured $1, 3, 2, 3, 4, 4, 5$ from left to right. Then, Heimdall could have taken the following journey:

- He began by walking right along bridge 1 (with colour 1),
- He then walked right along bridge 2 (with colour 3),
- He then walked left along bridge 2 (with colour 3),
- He then walked left along bridge 1 (with colour 1),
- He then walked right along bridge 1 (with colour 1),
- He then walked right along bridge 2 (with colour 3),
- He then walked right along bridge 3 (with colour 2),
- He then walked right along bridge 4 (with colour 3),
- He then walked right along bridge 5 (with colour 4),
- He then walked right along bridge 6 (with colour 4),
- He then walked right along bridge 7 (with colour 5), finishing his journey.

Loki could have taken the following journey:

- He began by walking right along bridge 1 (with colour 1),
- He then walked right along bridge 2 (with colour 3),
- He then walked right along bridge 3 (with colour 2),
- He then walked right along bridge 4 (with colour 3),
- He then walked right along bridge 5 (with colour 4),
- He then walked right along bridge 6 (with colour 4),
- He then walked left along bridge 6 (with colour 4),
- He then walked left along bridge 5 (with colour 4),
- He then walked left along bridge 4 (with colour 3),
- He then walked right along bridge 4 (with colour 3),
- He then walked right along bridge 5 (with colour 4),
- He then walked right along bridge 6 (with colour 4),
- He then walked right along bridge 7 (with colour 5),
- He then walked left along bridge 7 (with colour 5),
- He then walked right along bridge 7 (with colour 5), finishing his journey.

In Sample Input 2, there is no sequence that agrees with both Heimdall and Loki's notebooks.

Bongi

Stop what you are doing, a new very fun game has just been released: *Bongi*. The game is played using N balls. The ith ball has the positive integer a_i written on it.

You must divide the balls into K baskets, numbered 1 to K. Every ball must go into one of the baskets and every basket must contain at least one ball. The *value* of a basket is the sum of the integers on the balls in it. Your *score* is the geometric mean[1] of the values of the baskets.

Consider the following example with $K = 2$ baskets and $N = 5$ balls of value 12, 14, 16, 17 and 25. One possible solution is as follows:

- The first basket contains 16 and 17, and
- The second basket contains 25, 12 and 14.

The score of this solution is $(16 + 17)^{0.5} \cdot (25 + 12 + 14)^{0.5} \approx 41.02$.

You do not have to produce the maximum score possible, instead you are awarded points based on the score you are able to achieve.

This is an **output only** problem. You do not submit source code for this task. Instead, you are given a series of input files and must submit the corresponding output files.

Constraints

For all test cases:

- $1 \leq a_i \leq 10000$ for all i.
- $2 \leq K \leq N \leq 100$.

Input

- The first line of input contains the two integers N and K.
- The second line contains N integers $a_1, a_2, \ldots, a_N$.

Output

Output K lines, each containing a collection of integers, such that the ith line contains the values of the balls in the ith basket.

[1] The geometric mean of k values $a_1, a_2, ..., a_k$ is defined as $a_1^{1/k} \times a_2^{1/k} \times ... \times a_k^{1/k}$. For example, the geometric mean of 8, 9 and 10 is $8^{1/3} \times 9^{1/3} \times 10^{1/3} \approx 2 \times 2.080 \times 2.154 \approx 8.963$.

Scoring

If you do not put every ball into exactly one basket, and have each basket contain at least one ball, you will score 0% of the points for that test case.

Otherwise, let X be the score of your solution, B be the judges' best solution and $S = a_1 + a_2 + \cdots + a_N$. Then you will score

$$\min\left(100 \cdot \frac{10 + S/K - B}{10 + S/K - X}, 100\right)$$

percent of the points for that test case.

Note that it is possible to prove that $X \leq S/K$, thus the scoring function is well-defined.

Testcase	Points	N	K	Judges' best score
1	5	5	2	42
2	10	20	10	9564.453334019241
3	10	100	10	55237.69999809915
4	15	100	20	25324.1999731493
5	20	100	30	16980.161334096203
6	20	100	40	13886.860798648839
7	10	100	50	10086.367502580433
8	10	100	60	8392.505093895417

Submitting

To submit your output file for test case X, you must upload a file named `output_X.txt`. For example, for test case 4, the output file must be named `output_4.txt`.

You can choose to upload your output file for each test case individually, or together in a zip file.

Your score for this problem is the sum of your maximum scores for each test case, over all submissions you have made to this problem.

Sample Input 1

```
5 2
16 25 17 14 12
```

Sample Output 1

```
16 17
25 12 14
```

Explanation

The score of the solution is $X = (16 + 17)^{0.5} \cdot (25 + 12 + 14)^{0.5} \approx 41.02$. For this example, the best solution of the jury has score 42. Therefore, the percentage of points obtained by this output is

$$\approx 100 \cdot \frac{10 + 84/2 - 42}{10 + 84/2 - 41.02} \approx 91.11.$$

This example corresponds to the first test, which is worth 5 points. The number of points obtained when submitting this solution is 4.56/5.

Sample Code

You are provided with two files, `template.cpp` and `bongi.hpp`, to help you implement your solution.

The file `bongi.hpp` contains two functions that you may find useful:

```
double get_score(vector<vector<int> > solution);
```

This function returns the score of a solution. For example, if `solution = [[ 16, 17 ], [ 25, 12, 14 ]]`, then `get_score(solution)` returns (approximately) `41.02`.

```
double get_percentage(vector<vector<int> > solution);
```

This function returns the percentage of points that a solution would be given. For example, if `solution = [[ 16, 17 ], [ 25, 12, 14 ]]`, then `get_percentage(solution)` returns (approximately) `91.11`.

The file `template.cpp` contains an example usage of `get_percentage`. You can compile `template.cpp` using this command:

```
g++ -Wall -std=c++17 -O2 -o template template.cpp
```

Hedge Maze

Puss and Kitty must face the diabolical challenge of the *Down-Right Hedge Maze*. The maze is an $R \times C$ grid of cells with R rows (numbered 1 to R from top to bottom) and C columns (numbered 1 to C from left to right). The cell in the rth column and cth row is denoted (r, c). There are three types of cells, denoted by an uppercase character:

- `D`: It is only allowed to move downwards from this cell (if there is a cell there).
- `R`: It is only allowed to move rightwards from this cell (if there is a cell there).
- `B`: It is allowed to move downwards or rightwards from this cell (if there is a cell there).

It is possible to reach cell (R, C) from every cell in the maze.

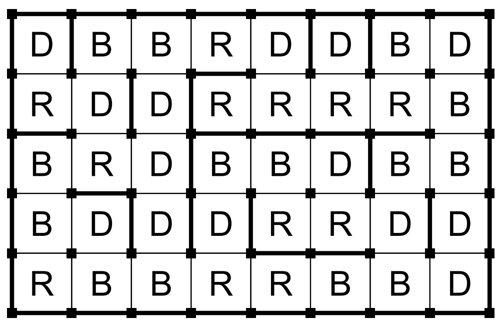

Figure 1: The maze from Sample Input 1. Thick black lines are drawn between cells where a move is disallowed.

To defeat the challenge, Puss and Kitty must answer Q queries. In each query, you are given four integers a_i, b_i, c_i and d_i and must answer the following question: If Puss starts in the cell (a_i, b_i) and Kitty starts in the cell (c_i, d_i), what is the the fewest total moves they must make to meet at a common cell? Puss and Kitty start in different cells.

Subtasks and Constraints

For all subtasks:

- $2 \leq R \leq 1000$
- $2 \leq C \leq 500\,000$
- $4 \leq R \times C \leq 1\,000\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq a_i \leq R$ and $1 \leq b_i \leq C$ for all i.
- $1 \leq c_i \leq R$ and $1 \leq d_i \leq C$ for all i.
- $(a_i, b_i) \neq (c_i, d_i)$ for all i.

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	9	$R, C \leq 20$
2	17	$(a_i, b_i) = (1, 1)$ for all i.
3	20	Each cell will be type `D` or `R`.
4	29	There is a sequence of moves from $(1, 1)$ to any cell in the maze.
5	25	No additional constraints.

Input

- The first line of input contains the three integers R, C, Q.
- R lines follow, each containing a string of C characters, describing the maze. The cth character on the rth line represents the type of the cell (r, c).
- Q lines follow, describing the queries. The ith line contains the four integers a_i, b_i, c_i and d_i.

Output

Output Q lines. On the ith line, print the answer to the ith query.

Sample Input 1

```
5 8 3
DBBRDDBD
RDDRRRRB
BRDBBDBB
BDDDRRDD
RBBRRBBD
1 2 3 7
3 3 1 1
3 6 4 4
```

Sample Output 1

```
9
4
7
```

Sample Input 2

```
4 4 4
BBRD
DRBB
BDRD
BBRB
2 1 3 3
1 3 2 2
3 3 1 3
4 1 3 2
```

Sample Output 2

```
7
4
4
2
```

Explanation

In Sample Input 1:

- For the first query, Puss and Kitty start in cell $(1, 2)$ and $(3, 7)$ respectively. They can meet in cell $(3, 8)$, requiring 8 and 1 moves respectively, for a total of 9 moves.
- For the second query, Puss and Kitty start in cell $(3, 3)$ and $(1, 1)$ respectively. They can meet in cell $(3, 3)$, requiring 0 and 4 moves respectively, for a total of 4 moves.
- For the third query, Puss and Kitty start in cell $(3, 6)$ and $(4, 4)$ respectively. They can meet at cell $(5, 7)$, requiring 3 and 4 moves respectively, for a total of 7 moves.

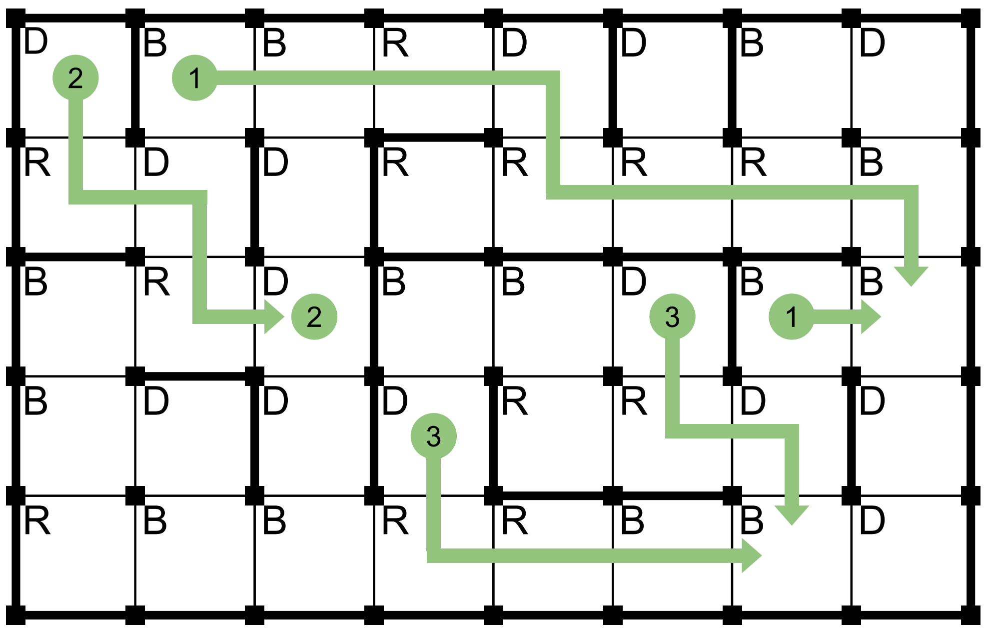

Figure 2: Sample Input 1

In Sample Input 2:

- For the first query, Puss and Kitty start in cell $(2, 1)$ and $(3, 3)$ respectively. They can meet in cell $(4, 4)$, requiring 5 and 2 moves respectively, for a total of 7 moves.
- For the second query, Puss and Kitty start in cell $(1, 3)$ and $(2, 2)$ respectively. They can meet in cell $(2, 4)$, requiring 2 and 2 moves respectively, for a total of 4 moves.
- For the third query, Puss and Kitty start in cell $(3, 3)$ and $(1, 3)$ respectively. They can meet at cell $(3, 4)$, requiring 1 and 3 moves respectively, for a total of 4 moves.
- For the fourth query, Puss and Kitty start in cell $(4, 1)$ and $(3, 2)$ respectively. They can meet at cell $(4, 2)$, requiring 1 and 1 move respectively, for a total of 2 moves.

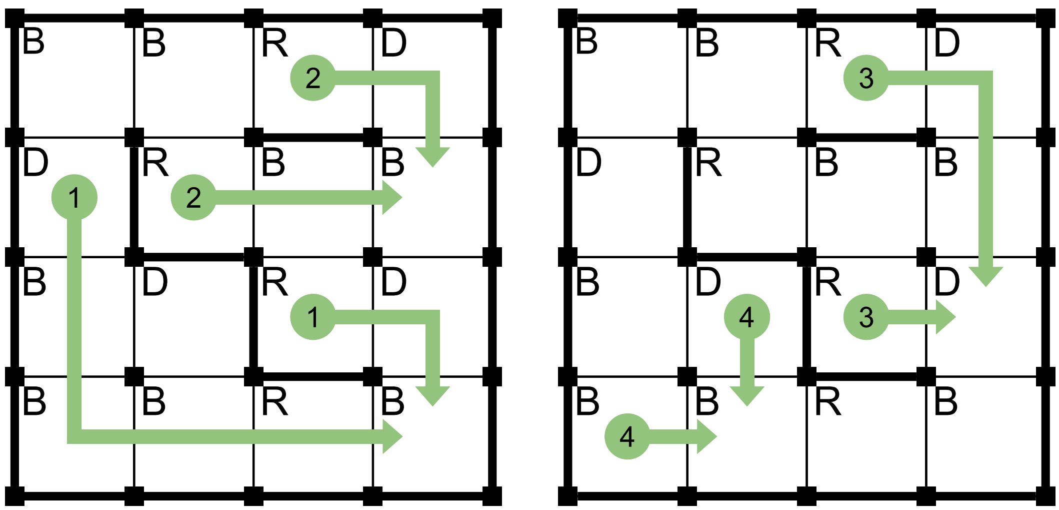

Figure 3: Sample Input 2. Queries 3 and 4 are shown on a separate copy of the maze for clarity.