Problem A. Inverse Inversions

Time Limit 1000 ms
Mem Limit 524288 kB

Your task is to create a permutation of numbers $1, 2, \ldots, n$ that has exactly k inversions.

An inversion is a pair (a, b) where $a < b$ and $p_a > p_b$ where p_i denotes the number at position i in the permutation.

Input

The only input line has two integers n and k.

Output

Print a line that contains the permutation. You can print any valid solution.

Constraints

- $1 \leq n \leq 10^6$
- $0 \leq k \leq \frac{n(n-1)}{2}$

Example

Input	Output
5 4	1 5 2 4 3

Problem B. Monotone Subsequences

Time Limit 1000 ms
Mem Limit 524288 kB

Your task is to create a permutation of numbers $1, 2, \ldots, n$ whose longest monotone subsequence has exactly k elements.

A monotone subsequence is either increasing or decreasing. For example, some monotone subsequences in $[2, 1, 4, 5, 3]$ are $[2, 4, 5]$ and $[4, 3]$.

Input

The first input line has an integer t: the number of tests.

After this, there are t lines. Each line has two integers n and k.

Output

For each test, print a line that contains the permutation. You can print any valid solution. If there are no solutions, print `IMPOSSIBLE`.

Constraints

- $1 \le t \le 1000$
- $1 \le k \le n \le 100$

Example

Input	Output
3 5 3 5 2 7 7	2 1 4 5 3 IMPOSSIBLE 1 2 3 4 5 6 7

Problem C. Third Permutation

Time Limit 1000 ms

Mem Limit 524288 kB

You are given two permutations a and b such that $a_i \neq b_i$ in every position. Create a third permutation c such that $a_i \neq c_i$ and $b_i \neq c_i$ in every position.

Input

The first line has an integer n: the permutation size.

The second line has n integers $a_1, a_2, \ldots, a_n$.

The third line has n integers $b_1, b_2, \ldots, b_n$.

Output

Print n integers $c_1, c_2, \ldots, c_n$. You can print any valid solution. If there are no solutions, print `IMPOSSIBLE`.

Constraints

- $2 \leq n \leq 10^5$

Example

Input	Output
5 1 3 2 5 4 4 1 3 2 5	3 2 5 4 1

Problem D. Permutation Prime Sums

Time Limit	1000 ms
Mem Limit	524288 kB

Given n, create two permutations a and b of size n such that $a_i + b_i$ is prime for $i = 1, 2, \ldots, n$.

Input

The only line has an integer n.

Output

Print two permutations. You can print any valid solution. If there are no solutions, print `IMPOSSIBLE`.

Constraints

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

Example

Input	Output
5	2 1 3 5 4 5 1 4 2 3

Explanation: The sums are $2 + 5 = 7$, $1 + 1 = 2$, $3 + 4 = 7$, $5 + 2 = 7$ and $4 + 3 = 7$ which all are primes.

Problem E. Chess Tournament

Time Limit 1000 ms

Mem Limit 524288 kB

There will be a chess tournament of n players. Each player has announced the number of games they want to play.

Each pair of players can play at most one game. Your task is to determine which games will be played so that everybody will be happy.

Input

The first input line has an integer n: the number of players. The players are numbered $1, 2, \ldots, n$.

The next line has n integers $x_1, x_2, \ldots, x_n$: for each player, the number of games they want to play.

Output

First print an integer k: the number of games. Then, print k lines describing the games. You can print any valid solution.

If there are no solutions, print "IMPOSSIBLE".

Constraints

- $1 \leq n \leq 10^5$
- $\sum_{i=1}^{n} x_i \leq 2 \cdot 10^5$

Example

Input	Output
5 1 3 2 0 2	4 1 2 2 3 2 5 3 5

Problem F. Distinct Sums Grid

Time Limit 1000 ms
Mem Limit 524288 kB

Create an $n \times n$ grid that fulfills the following requirements:

1. Each integer $1 \ldots n$ appears n times in the grid.
2. If we create a set that consists of all sums in rows and columns, there are $2n$ distinct values.

Input

The only line has an integer n.

Output

Print a grid that fulfills the requirements. You can print any valid solution. If there are no solutions, print `IMPOSSIBLE`.

Constraints

- $1 \leq n \leq 1000$

Example

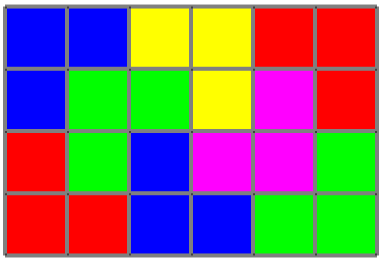
Input	Output
5	2 3 1 1 1 1 5 5 3 3 2 3 5 2 4 5 4 5 4 1 2 3 4 4 2

Explanation: Each integer $1 \ldots 5$ appears 5 times, and the sums in rows and columns are $\{8, 11, 12, 14, 15, 16, 17, 18, 19, 20\}$.

Problem G. Filling Trominos

Time Limit 1000 ms
Mem Limit 524288 kB

Your task is to fill an $n \times m$ grid using L-trominos (three squares that have an L-shape). For example, here is one way to fill a 4×6 grid:



Input

The first input line has an integer t: the number of tests.

After that, there are t lines that describe the tests. Each line has two integers n and m.

Output

For each test, print YES if there is a solution, and NO otherwise.

If there is a solution, also print n lines that each contain m letters between A–Z. Adjacent squares must have the same letter exactly when they belong to the same tromino. You can print any valid solution.

Constraints

- $1 \leq t \leq 100$
- $1 \leq n, m \leq 100$

Example

Input	Output
2 4 6 4 7	YES AADDBB ACCDEB BCAEEC BBAACC NO

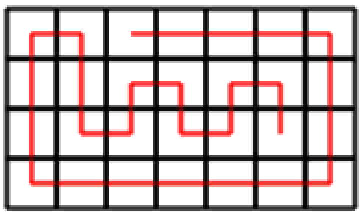
Problem H. Grid Path Construction

Time Limit 1000 ms

Mem Limit 524288 kB

Given an $n \times m$ grid and two squares $a = (y_1, x_1)$ and $b = (y_2, x_2)$, create a path from a to b that visits each square exactly once.

For example, here is a path from $a = (1, 3)$ to $b = (3, 6)$ in a 4×7 grid:



Input

The first input line has an integer t: the number of tests.

After this, there are t lines that describe the tests. Each line has six integers n, m, y_1, x_1, y_2 and x_2.

In all tests $1 \leq y_1, y_2 \leq n$ and $1 \leq x_1, x_2 \leq m$. In addition, $y_1 \neq y_2$ or $x_1 \neq x_2$.

Output

Print YES, if it is possible to construct a path, and NO otherwise.

If there is a path, also print its description which consists of characters `U` (up), `D` (down), `L` (left) and `R` (right). If there are several paths, you can print any of them.

Constraints

- $1 \leq t \leq 100$
- $1 \leq n \leq 50$
- $1 \leq m \leq 50$

Example

Input	Output
5 1 3 1 1 1 3 1 3 1 2 1 3 2 2 1 1 2 2 2 2 1 1 2 1 4 7 1 3 3 6	YES RR NO NO YES RDL YES RRRRDDDLLLLLLUUURDDRURDRURD