Cursedtablet

A long time ago in Uzbekistan, a boy named Alisher found a strange stone tablet under an apricot tree. It had **two rows of numbers** on it:

- The **first row** shows how things are right now - maybe bad, maybe okay.
- The **second row** shows how things *should* be - happy, peaceful, and perfect.

Your job is to help Alisher **turn the first row into the second row**, using special magical moves.

Magic Moves You Can Use

Alisher can use these two types of moves:

- $1\ l\ r\ x$ – Choose **exactly K numbers in a row**, starting from position l until position r, and **add the number x** (positive or negative) to each of them. Note that r should always be equal to $l + K - 1$ (e.g: If $l = 2$ and $K = 3$, then a magic move can be $124x$: you add x to positions 2, 3, and 4.)
- 2 – Flip the whole row backward! (The first becomes last, the second becomes second-to-last, etc.)

Important Rules

- Alisher must **not** let any number become bigger than 10^{16} or smaller than -10^{16} after any move.
- Some puzzles are **impossible to solve** — if you can't fix the first row, print -1.
- If it's possible, find the **smallest number of moves** and show which moves to do.

Input

The first line has a number T — the number of puzzles.

For each puzzle:

- First line: two numbers N and K — the number of symbols and the size of the group Alisher can change in one move.
- Second line: N numbers — the current state (first row).
- Third line: N numbers — the goal state (second row).

Output

For each puzzle:

- If it's **not possible** to fix the first row, print -1.
- Otherwise:
  - First, print a number — how many magic moves Alisher needs.
  - Then print each move on its own line:
    - Either $1\ l\ r\ x$
    - Or 2

Subtasks

You will get **100% of the score for subtask** if:

- You print the **minimum number of moves** - mn, and
- Your operations **correctly** turn the first row into the second row in every test case.

You will get **50% of the score for a subtask** if:

- You print mn and the operations are in the **correct format**, but
- The operations **don’t actually** make the first row equal to the second.

So even if you can't solve a test case, printing any valid mn and well-formed operations can still earn you partial credit.

1. (20 points) $a_1 = a_2 = \ldots = a_N$, $b_1 = b_2 = \ldots = b_N$
2. (18 points) $K = 1$
3. (12 points) $K = N$
4. (50 points) No additional constraints

Examples

Example 1

Input

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

Output

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