Handcrafted Gift (gift)

Adam is making Bob a hand-crafted necklace as a gift. A necklace consists of n beads, numbered 0 to $n-1$ from left to right. Each bead can either be **red** or **blue** in colour. Bob has sent Adam a list of r requirements for the necklace. The ith requirement ($0 \leq i < r$) states that the beads from positions $a[i]$ to $b[i]$ inclusive should have $x[i]$ unique colours.

Help Adam find a possible configuration of beads that satisfies all of Bob's requirements, or determine that it is impossible.

Implementation Details

You should implement the following procedure:

```
int construct(int n, int r, int[] a, int[] b, int[] x)
```

- n: number of beads.
- r: number of requirements.
- a: an array of length r, the starting position of each requirement.
- b: an array of length r, the ending position of each requirement.
- x: an array of length r, the number of unique colours for each requirement.
- This procedure will be called exactly once.
- If a construction is possible, this procedure should make exactly one call to `craft` to report the construction, following which it should return 1.
- Otherwise, the procedure should return 0 without making any calls to `craft`.

Your program should call the following procedure to report the construction:

```
void craft(string s)
```

- s, a string of length n, with $s[i]$ equal to 'R' if the ith bead is red, or 'B' if it is blue.

Examples

Example 1

Consider the following call:

```
construct(4, 2, [0, 2], [2, 3], [1, 2])
```

This means that there are a total of 4 beads and 2 requirements as follows:

- positions 0 to 2 should have 1 unique colour,
- positions 2 to 3 should have 2 unique colours.

This can be achieved by colouring beads 0 to 2 red, and bead 3 blue.

Therefore, the `construct` procedure should make the following call:

- `craft("RRRB")`

It should then return 1.

In this case, there are multiple constructions that fit the requirements, all of which would be considered correct.

Example 2

Consider the following call:

```
construct(3, 3, [0, 1, 0], [1, 2, 2], [1, 1, 2])
```

This means that there are a total of 3 beads and 3 requirements as follows:

- positions 0 to 1 should have 1 unique colour,
- positions 1 to 2 should have 1 unique colour,
- positions 0 to 2 should have 2 unique colours.

In this case, there are no possible configuration of beads that satisfy all the requirements.

As such, the `construct` procedure should return 0 without making any call to `craft`.

Constraints

- $1 \leq n, r \leq 500\,000$
- $0 \leq a[i] \leq b[i] \leq n - 1$ (for all $0 \leq i \leq r - 1$)
- $1 \leq x[i] \leq 2$ (for all $0 \leq i \leq r - 1$)

Subtasks

1. (10 points) $x[i] = 1$ (for all $0 \leq i \leq r - 1$)
2. (15 points) $x[i] = 2$ (for all $0 \leq i \leq r - 1$)
3. (20 points) $1 \leq n, r \leq 18$
4. (25 points) $1 \leq n, r \leq 2000$

5. (30 points) No additional constraints.

Sample grader

The sample grader reads the input in the following format:

- line 1: n r
- line $2 + i$ ($0 \leq i \leq r - 1$): $a[i]$ $b[i]$ $x[i]$

The sample grader prints your answers in the following format:

- line 1: the return value of `construct`.
- line 2: s
- If the return value of `construct` is 0, s will not be printed.