Lab 4 A

Problem 1: Smallest Set Cover

An interval is a pair of positive integers [a, b] with a ≤ b. It is meant to denote the set of integers that lie between the values a and b. For example [3,5] denotes the set {3,4,5} while the interval [3, 3] denotes the set {3}.

We say that an interval [a, b] is covered by an integer i, if i belongs to the set defined by [a, b]. For example interval [3, 5] is covered by 3 and so is the interval [3, 3].

Given a set of intervals I, and a set of integers S we say that I is covered by S if for each interval [a, b] in I there is an integer i in S such that [a, b] is covered by i. For example, the set {[3, 5], [3, 3]} is covered by the set {3}. The set of intervals {[6, 9], [3, 5], [4, 8]} is covered by the set {4, 5, 8}. It is also covered by the set {4, 7}.

We would like to compute, for any set of intervals I, the size of the smallest set S that covers it. You can check that for the set of intervals {[6, 9], [3, 5], [4, 8]} the answer is 2 while for the set of intervals {[3, 5], [3, 3]} the answer is 1.

Input format:

- The first line contains a single integer N, giving the number of intervals in the input.
- This is followed by N lines, each containing two integers separated by a space describing an interval, with the first integer guaranteed to be less than or equal to the second integer.

Output format:

- Output a single integer giving the size of the smallest set of integers that covers the given set of intervals.

Constraints:

- `1 ≤ N ≤ 1000` (Number of intervals)
- `1 ≤ a ≤ b ≤ 1,000,000,000` (Bounds of each interval)

Example:

Sample Input:

```
3
6 9
3 5
4 8
```

Sample Output:

```
2
```

Explanation:

The set of intervals is {[6, 9], [3, 5], [4, 8]}. The smallest set of integers that covers these intervals is {4, 8}. Both integers 4 and 8 cover all the intervals:

- Interval [6, 9] is covered by 8.
- Interval [3, 5] is covered by 4.
- Interval [4, 8] is covered by both 4 and 8.

Hence, the size of the smallest set is 2.

Problem 2: Zainab's Caps Lock Conundrum

Zainab, our lovable tech-challenged friend, was trying to set up her email account but didn't realize her Caps Lock was on. As a result, both her email ID and password ended up in the wrong case. While she fixed her password she didn't bother fixing her email (surprisingly, she is smart enough to know that email IDs are case insensitive).

Please help little Zainab by correcting her password while keeping her email ID untouched.

Input:

- The first line contains three integers: N_t, N_e, and N_p. Here, N_t is the total number of characters (email ID + space + password), N_e is the number of characters in the email ID, and N_p is the number of characters in the password.
- Second line contains the incorrectly typed email ID and password, separated by a

space.

Output:

- Print the corrected email ID (which remains unchanged) and the corrected password. The password should be converted to the correct case while keeping the email ID in its original form.

Constraints:

- Both the email ID and password contain only alphabets, digits, `@`, `_`, and `.`
- $N_t = N_e + 1 + N_p$.
- $0 \leq N_t \leq 1000$

Example:

Input:

```
31 20 10
        TESTEMAIL@DOMAIN.COM aBCDefghIJ
```

Copy

Output:

```
TESTEMAIL@DOMAIN.COM AbcdEFGHij
```

Explanation:

Here, we correct the password by converting `aBCDefghIJ` to `AbcdEFGHij` while keeping the email ID unchanged.

Problem 3: Bitville Street Light Analyzer

In the small town of Bitville, the mayor has recently implemented a new binary communication system for the town's street lights. Each street light is represented by a binary value, either 0 (off) or 1 (on). The street lights are arranged in a line, and the system records the current state of all the lights in an array.

The mayor wants to know the energy consumption pattern in certain sections of the town by

analyzing the street lights. For this, he needs to find out whether the number of street lights that are on (represented by 1s) in a given range is odd or even.

Your task is to help the mayor by answering multiple queries. Each query will ask you to analyze a specific range of street lights, from position `l` to position `r`, inclusive. For each query, you should determine if the number of 1s in the given range is odd or even.

Return `ODD` or `EVEN` based on the number of on lights in the given range

Input:

- First line containing two space separated integers,number of lights on the street n,number of queries q
- Second line containing n spaced integer representing the state of lights, say array `arr`
- next q lines containing two spaced integers l and r

Output:

- q lines of output, `ODD` or `EVEN` based on the number of lights in the range given in the query

Constraints:

- `1<=N<=1e5`
- `1<=Q<=1e5`
- `0<=arr[i]<=1`

Good luck!

Input:

```
5 3
1 0 1 1 0
1 3
2 5
1 5
```

Copy

Output:

```
EVEN
EVEN
ODD
```

Explanation:

for first query the range is from 1 to 3, which is `1 0 1` which has even number of ones so output is `EVEN`

for third query the range is from 1 to 5, which is `1 0 1 1 0` which has odd number of ones so output is `ODD`

Input:

```
1 1
0
1 1
```

Copy

Output:

```
EVEN
```

Explanation:

for first query the range is from 1 to 1, which is `0` which has zero number of ones so output is `EVEN`

Submission Guidelines

- Do not rename any files given in the handout. Only write the code in the specified C files in the respective directories.