

# January 2024 CSE 102

## Offline 2: Array

Total Mark: 10 + 10 = 20

### Problem 1: Count the pairs

You are given two arrays of size **N**, each containing a permutation of numbers from **1** to **N**. Your task is to count how many pairs of numbers (**a**, **b**) are in the same order in both arrays. Specifically, **a** should come before **b** in both arrays.

#### Input

1. A single integer **N**, representing the size of the two arrays.
2. The next line contains **N** integers, representing the first permutation of numbers from 1 to **N**.
3. The following line contains **N** integers, representing the second permutation of numbers from 1 to **N**.

Both arrays are guaranteed to be valid permutations of numbers from **1** to **N**.

Input	Output
4 4 1 3 2 1 4 2 3	4 /** Explanation: The pairs are (4, 3), (4, 2), (1, 3), (1, 2) **/
5 1 2 3 4 5 1 2 3 4 5	10 /** Explanation: All pairs are in same order in both arrays **/

In the first input case, observe that 4 occurs before 3 in both arrays, so (4, 3) is a valid pair. Similarly, (4, 2), (1, 3), and (1, 2) are also valid.

## Problem 2: Card Passing Game

Your task is to simulate a game where children are standing in a circle, each holding a numbered card. **Children's positions are fixed starting from 1**. Based on a series of game instructions, the children will pass their cards either to the right or to the left. After all the instructions are completed, the child holding the highest numbered card will be declared the winner. The instruction is just a number.

- If the instruction is a positive number  $k$ , each child passes their card to the right  $k$  times.
- If an instruction is a negative number  $k$ , each child passes their card to the left  $k$  times.

After each instruction, the current order of the cards must be printed. At the end of all passes, the child holding the highest numbered card wins the game. Finally, the position of the winner must be printed.

**You have to do all the operations on a single array. You can define extra variables but not arrays.**

### Input:

- The first input is an integer  $n$ , representing the number of children (or the number of cards).
- $n$  integers, where each integer corresponds to the card number initially held by the child in that position.
- The third input is an integer  $m$ , representing the number of instructions.
- $m$  integers, where:
  1. If the number is positive, pass the cards to the right by that number of positions.
  2. If the number is negative, pass the cards to the left by that number of positions

**Output:**

You have to print the current order of the cards after each instruction. After processing all the instructions, output the final arrangement of the cards and output the winner.

**Follow the I/O format given below.**

Input	Output
5 1 2 3 4 5 3 2 -1 3	Initial order: 1 2 3 4 5 After instruction 1: 4 5 1 2 3 After instruction 2: 5 1 2 3 4 After instruction 3: 2 3 4 5 1 Winner: Child 4
6 10 20 60 50 40 30 5 3 -2 1 -4 2	Initial order: 10 20 60 50 40 30 After instruction 1: 50 40 30 10 20 60 After instruction 2: 30 10 20 60 50 40 After instruction 3: 40 30 10 20 60 50 After instruction 4: 60 50 40 30 10 20 After instruction 5: 10 20 60 50 40 30 Winner: Child 3

**Explanation of test case 1:**

- After the 1st instruction(2), the sequence shifted 2 positions to the right circularly.
- After the 2nd instruction(-1), the sequence shifted 1 position to the left circularly
- After the 3rd instruction(3), the sequence shifted 3 positions to the right circularly.
- Finally, the highest card number (5) is at 4th position. So, the winner is child 4.

**Explanation of test case 2:**

- After the 1st instruction(3), the sequence shifted 3 positions to the right circularly.
- After the 2nd instruction(-2), the sequence shifted 2 positions to the left circularly.
- After the 3rd instruction(1), the sequence shifted 1 position to the right circularly.

- After the 4th instruction(-4), the sequence shifted 4 positions to the left circularly.
- After the 5th instruction(2), the sequence shifted 2 positions to the right circularly.
- Finally, the highest card number(60) is at 3rd position. So, the winner is child 3.

**Submission Guideline:**

1. Go to a drive except C drive.
2. Create a folder according to your roll number. Ex- 2305xxx
3. Open up the folder and create two files there. Ex- 2305xxx-1.c, 2305xxx-2.c.
4. Place all the code inside the two .c files.
5. Zip the folder and submit it in the moodle.