**Data Structure Lab**

**ODD 2022**

**Lab Test-1**

**Time: 50 min**

**Marks: 20**

**Instructions**:

1. Submit a pdf file having code and output screenshots
2. FilenameshouldbeRollNo\_Name\_LabTest1\_DS.pdf
3. Output should be at least 3 unique test cases
4. Output Screenshot should NOT be cropped at all

**Set 1 (ODD System No)**

**Q1 (CO1)**The school cafeteria offers **circular** and **square** sandwiches at lunch break, referred to by numbers **0** and **1** respectively. All students stand in a queue. Each student either prefers square or circular sandwiches.

The number of sandwiches in the cafeteria is equal to the number of students. The sandwiches are placed in a stack. At each step:

* If the student at the front of the queue prefers the sandwich on the top of the stack, they will take it and leave the queue.
* Otherwise, they will leave it and go to the queue's end.

This continues until none of the queue students want to take the top sandwich and are thus unable to eat.

You are given two integer arrays students and sandwiches where sandwiches[i] is the type of the i​​​​​​th sandwich in the stack (i = 0 is the top of the stack) and students[j] is the preference of the j​​​​​​th student in the initial queue (j = 0 is the front of the queue). Return the number of students that are unable to eat.

**Example 1:**

Input: students = [1,1,0,0], sandwiches = [0,1,0,1]

Output: 0

**Explanation:**

- Front student leaves the top sandwich and returns to the end of the line making students = [1,0,0,1].

- Front student leaves the top sandwich and returns to the end of the line making students = [0,0,1,1].

- Front student takes the top sandwich and leaves the line making students = [0,1,1] and sandwiches = [1,0,1].

- Front student leaves the top sandwich and returns to the end of the line making students = [1,1,0].

- Front student takes the top sandwich and leaves the line making students = [1,0] and sandwiches = [0,1].

- Front student leaves the top sandwich and returns to the end of the line making students = [0,1].

- Front student takes the top sandwich and leaves the line making students = [1] and sandwiches = [1].

- Front student takes the top sandwich and leaves the line making students = [] and sandwiches = [].

Hence all students are able to eat.

**Example 2:**

Input: students = [1,1,1,0,0,1], sandwiches = [1,0,0,0,1,1]

**Output: 3**

**Q2 (CO2)**Given a sorted integer array **arr**, two integers **k** and **x**, return the **k** closest integers to **x** in the array. The result should also be sorted in ascending order.

An integer a is closer to x than an integer b if:

|a - x| < |b - x|, or

|a - x| == |b - x| and a < b

**Example 1:**

**Input**: arr = [1,2,3,4,5], k = 4, x = 3

**Output**: [1,2,3,4]

**Example 2:**

**Input**: arr = [1,2,3,4,5], k = 4, x = -1

**Output**: [1,2,3,4]

**Set 2 (Even System No)**

**Q1 (CO1)**Make a class **Stack** with following operations:

Push(int x): to insert data x into stack

Pop(): to remove data from the top of the stack

Isempty():to check if stack is empty

Now, implement the functionality of **Queue** using two stack objects. The prototype of the queue class would be as follows:

class Queue {

    public:

    Stack S1, S2;

     void enqueue(int x);

      int dequeue();

}

Implement all the required operations to perform the following operations:

enqueue(10);

enqueue(100);

enqueue(1000);

display();

dequeue();

display();

**Q2 (CO2)** In a cricket tournament, n  matches are supposed to be played between Bob and James. You are given two arrays A and B. Array A represents the energy levels of Bob and array B represents the energy levels of James. The size of both the arrays is the same as n.

Now, n matches will be played between Bob and James, and for the ith match the energy level at ith index is compared of both. The winner of the ith match will be the one whose energy level at that index is higher.

If Bob wins in the ith match, then the energy level difference (between the ith level of both) will be added to his score else his score remains the same i.e., on losing, Bob's points will not decrease.

Bob wants to maximize his score by changing his energy levels, that is, Bob can make a permutation of his array.  Help Bob in knowing the maximum score he can get.

**Input format**

* The first line of each test case contains an integer n denoting the number of matches.
* Each of the next two lines contains n integers representing energy levels.

**Output format**

* Print a single integer denoting the maximum score.

**For example:**

**Sample Input**

5

1 2 3 4 5

1 2 3 4 5

**Sample Output**

6

**Explanation**

Bob will permute his array to {5, 4, 3, 2, 1}. So for the first and second levels, Bob will get a score of 4, 2 respectively, and for the levels, Bob won't get any positive score, So his total maximum score is 4+2 = 6.