M23CS1.304 Data Structures and Algorithms for Problem Solving

Assignment 0

Deadline: 11:59 pm. August 6, 2023

Important Points:

1. Only C++ is allowed.
2. **Directory Structure:**

   2023201001_A0
   |_____2023201001_A0_Q1.cpp
   |_____2023201001_A0_Q2.cpp
   Replace your roll number in place of 2023201001
3. **Submission Format:** Follow the above mentioned directory structure and zip the RollNo_A0 folder and submit RollNo_A0.zip on Moodle.

   Note: All submissions which are not in the specified format or submitted after the deadline will be awarded **0** in the assignment.
4. C++ STL is **not allowed** for any of the questions unless specified otherwise in the question. So "#include <bits/stdc++.h>" is not allowed.
5. You can ask queries by posting on Moodle.

Any case of plagiarism will lead to a 0 in the assignment or "F" in the course.

1. Anagram Check

Given a set of 2 words, print “YES” if they are anagrams and “NO” if not.

Problem Constraints:

$1 <= T <= 10$

$3 <= n <= 10^5$

Where T is the number of testcases.

All characters are lowercase alphabets.

Input Format:

The first line contains an integer T denoting the number of testcases.

The remaining T lines contain 2 space-separated strings.

Output Format:

Print "YES” or “NO” for each pair.

Sample Input / Output:

Input 1:

2

hello heoll

abc abe

Output 1:

YES

NO

2. Jumping Around

Consider a circular array of size N. Given a starting index S and an array of jumps J, print the values that the frog jumping around the array can reach.

Problem Constraints:

$1 <= N <= 10^6$

$0 <= S < N$

$0 <= J.length <= 10^6$

$0 <= J[I] <= 10^9$

Input Format:

The first line contains an integer N denoting the size of circular array, integer S denoting the starting index of frog and integer J denoting the size of jumps array.

The second line contains N elements for the array.

The third line contains J elements for the jumps array.

Output Format:

Print J + 1 numbers which will be the value of element of array on which frog has landed

Sample Input / Output:

Input 1:

5 3 4

1 8 5 7 9

9 13 3 0

Output 1:

7 5 1 7 7

Explanation 1:

The value at starting index (3) is 7, after that the frog jumps forward 9 elements in the circular array, reaching the value 5. Then, the frog jumps forward 13 elements and reaches the value 1. This is repeated until all the jumps are exhausted.