**100 Days of Code**

**DAY 1:**

**Longest Increasing Subsequence**

Given a sequence, find the length of the longest increasing subsequence from a given sequence .  
The longest increasing subsequence means to find a subsequence of a given sequence in which the subsequence's elements are in sorted order, lowest  
to highest, and in which the subsequence is as long as possible. This subsequence is not necessarily contiguous, or unique.

**Note:** Duplicate numbers are not counted as increasing subsequence.

For example:  
 LIS for { 10, 22, 9, 33, 21, 50, 41, 60, 80 } is {10, 22, 33, 50, 60, 80}.

# K’th Smallest/Largest Element in Unsorted Array

Given an array and a number k where k is smaller than size of array, we need to find the k’th smallest element in the given array. It is given that ll array elements are distinct.

Input: arr[] = {7, 10, 4, 3, 20, 15}

k = 3

Output: 7

Given an array A[] and a number x, check for pair in A[] with sum as x

Write a program that, given an array A[] of n numbers and another number x, determines whether or not there exist two elements in S whose sum is exactly x.