TUTORIAL 3

1. If A^2 mod C = **(A * A)** mod C = (**(A mod C)** * **(A mod C)**) mod C then calculate 7^{256} mod 13.

2. Write a program in any programming language to find the leader:

Leaders in an array

Write a program to print all the LEADERS in the array. An element is leader if it is greater than all the elements to its right side. And the rightmost element is always a leader. For example int the array {16, 17, 4, 3, 5, 2}, leaders are 17, 5 and 2. Let the input array be arr[] and size of the array be *size*.

Also calculate the complexity of the code.

3. Write a program in any language to find the number of inversion.

Inversion Count for an array indicates – how far (or close) the array is from being sorted. If array is already sorted then inversion count is 0. If array is sorted in reverse order that inversion count is the maximum.

Formally speaking, two elements a[i] and a[j] form an inversion if a[i] > a[j] and i < j.

Example:

The sequence 2, 4, 1, 3, 5 has three inversions (2, 1), (4, 1), (4, 3).

Use Bubble Sort, Insertion Sort and Selection sort for the same.

4. An Array of integers is given, both +ve and -ve. You need to find the two elements such that their sum is closest to zero. Write code in any programming language for the same.

5. Find the largest and second largest element with its index in a given array. The runtime complexity of this code should be linear.

6. Write a function to find if a given integer x appears more than n/2 times in a sorted array of n integers.

Basically, we need to write a function say is Majority() that takes an array (arr[]), array's size (n) and a number to be searched (x) as parameters and returns true if x is a majority element (present more than n/2 times).

Examples:

Input: arr[] = {1, 2, 3, 3, 3, 3, 10}, x = 3

Output: True (x appears more than n/2 times in the given array)

Input: arr[] = {1, 1, 2, 4, 4, 4, 6, 6}, x = 4

Output: False (x doesn't appear more than n/2 times in the given array)

Input: arr[] = {1, 1, 1, 2, 2}, x = 1

Output: True (x appears more than n/2 times in the given array)

7. You are given an array of 0s and 1s in random order. Segregate 0s on left side and 1s on right side of the array. Traverse array only once.

}

Input array = [0, 1, 0, 1, 0, 0, 1, 1, 1, 0]

Output array = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]

8. Given two sorted arrays, find their union and intersection.

Example:

Input : arr1[] = {1, 3, 4, 5, 7}

arr2[] = {2, 3, 5, 6}

Output : Union : {1, 2, 3, 4, 5, 6, 7}

Intersection : {3, 5}

Input : arr1[] = {2, 5, 6}

arr2[] = {4, 6, 8, 10}

Output : Union : {2, 4, 5, 6, 8, 10}

Intersection : {6}

9. A sorted array has n elements, how many searches M are going to be made for Binary search, when

   n= 1000, M = 10
   n = 16,000, M = 14
   n = 64000, M =16
   n = 1000000, M = 20
   n = 900, M =10
   n = 5000, M = 13