Section 14

PDS Lab **Lab - 7** **04.01.2023**

Instructions:

Give sufficient comment against each statement in your program.
You should save each program with the file name as per the convention against each problem.
There is a partial credit even if your program does not run successfully for all the test cases as mentioned in the problem.

Problem 1:

Write a program that will insert a new value to an already existing sorted array such that the ordering is maintained. You can assume the following:

a) The array is to be maintained in ascending order, for example, 1, 2, 3, …
b) An input array is given where all elements are already in sorted order.

User Input: Number of elements in the array, say *n*, *n* elements in the array in sorted order, the new value to be inserted.
Output: The input array, the output array

Test cases:

#	INPUT	OUTPUT

1	A[n]=5 10 15 20 25
	m=17
		Output Array: 5 10 15 17 20 25

2	A[n]=5 10 15 20 25
	m=30
		Output Array: 5 10 15 20 25 30

3	A[n]=2 4 6 8 10
	m=5
		Output Array: 2 4 5 6 8 10

4	A[n]=2 4 6 8 10
	m=0
		Output Array: 0 2 4 6 8 10

[Time: 25 Minutes] [20]

Problem 2:

An array contains *n* elements of integer type. The array may contain duplicate number. Read the array from the user as input. Then the user will further provide a number, say *m*. Your program should output all the numbers that appear *m* number of times.
(Assume maximum array length = 100).

Test cases:

#	INPUT	OUTPUT

1	n=1 2 3 3 4 5 6 6 7 8
	m=2
		3 6 appears 2 times

2	n=1 2 3 4 5 6 7 8 9	NO number appears 2 times

	m=2	

3	n=1 2 3 4 5 6 7 8 9
	m=1
		1 2 3 4 5 6 7 8 9 appears 1 times

4	n=1 1 1 1 1 1 1 1 1 1
	m=10
		1 appears 10 times

[Time: 25 Minutes] [20]

Problem 3:

Write a program to check if two given strings are Anagram. Two strings are called anagrams of each other, if they contain the same characters but maybe in different order. For example, “silent” and “listen”.

Write a program such that your program reads two strings from the user as input and check if they are anagram.

(Assume only single words(without spaces) will be given as input)

Test cases:

#	INPUT	OUTPUT
1	mary army	Anagram
2	stop pots	Anagram
3	PDSLAB PDSLAB	Anagram
4	cat ate	NOT Anagram

[Time: 25 Minutes] [20]

Problem 4:

Find the largest number possible from a given set of numbers where the numbers can be appended ed to each other so that the appended number becomes the largest number.

Write a program to read, say *n* ($n \geq 2$) numbers from the keyboard. Next, your program should print the largest number on the appending of input numbers.

Test cases:

#	INPUT	OUTPUT
1	1 1 1 1	1111
2	1 1 2 3	3211
3	10 68 75 7 21 12	77568211210 (7 75 68 21 12 10)
4	10 68 97 9 21 12	99768211210 (9 97 68 21 12 10)

[Time: 25 Minutes] [20]

Problem 5:

Consider *a[ ]* is an array of integer values. An element *a[i]* is said to be in order if it is not smaller than or equal to *a[0]*, *a[1]*, ..., *a[i-1]* and not larger than or equal to *a[i+1]*, *a[i+2]*, ..., *a[n-1]*.

Write a program that takes as input an integer array *a[10]* that stores *n* ($0 < n \leq 10$) elements from the keyboard and finds how many elements are “Out of order”.

Test cases:

#	INPUT	OUTPUT
1	5 7 8 10	Out of order: 0
2	5 5 7 5	Out of order: 2
3	2 4 2 2 4 2 2	Out of order: 6
4	1 1 1 1 1 1 1 1 1 1	Out of order: 0

[Time: 25 Minutes] [20]

-----*-----