Lab Manual

Course Title : Design and Analysis of Algorithms

Week 10:

I. Given a list of activities with their starting time and finishing time. Your goal is to select maximum number of activities that can be performed by a single person such that selected activities must be non-conflicting. Any activity is said to be non-conflicting if starting time of an activity is greater than or equal to the finishing time of the other activity. Assume that a person can only work on a single activity at a time.

Input Format:
First line of input will take number of activities N.
Second line will take N space-separated values defining starting time for all the N activities.
Third line of input will take N space-separated values defining finishing time for all the N activities.

Output Format:
Output will be the number of non-conflicting activities and the list of selected activities.

Sample I/O Problem I:

Input:	**Output:**
10 1 3 0 5 3 5 8 8 2 12 4 5 6 7 9 9 11 12 14 16	No. of non-conflicting activities: 4 List of selected activities: 1, 4, 7, 10

II. Given a long list of tasks. Each task takes specific time to accomplish it and each task has a deadline associated with it. You have to design an algorithm and implement it using a program to find maximum number of tasks that can be completed without crossing their deadlines and also find list of selected tasks.

Input Format:
First line will give total number of tasks n.
Second line of input will give n space-separated elements of array representing time taken by each task.
Third line of input will give n space-separated elements of array representing deadline associated with each task.

Output Format:
Output will be the total number of maximum tasks that can be completed.

Sample I/O Problem II:

Input:	**Output:**
7 2 1 3 2 2 2 1 2 3 8 6 2 5 3	Max number of tasks = 4 Selected task numbers : 1, 2, 3, 6

III. Given an unsorted array of elements, design an algorithm and implement it using a program to find whether majority element exists or not. Also find median of the array. A majority element is an element that appears more than n/2 times, where n is the size of array.

Input Format:
First line of input will give size n of array.

Second line of input will take n space-separated elements of array.

Output Format:
First line of output will be '**yes**' if majority element exists, otherwise print '**no**'.
Second line of output will print median of the array.

Sample I/O Problem III:

Input:	**Output:**
9 4 4 2 3 2 2 3 2 2	yes 2