**Problem A**

You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.

**Sample Input/Output**

Consider the following 3 activities sorted by by finish time.  
 start[] = {10, 12, 20};  
 finish[] = {20, 25, 30};  
A person can perform at most two activities. The maximum set of activities that can be executed is {0, 2} [ These are indexes in start[] and finish[] ]  
  
Consider the following 6 activities sorted by by finish time.  
 start[] = {1, 3, 0, 5, 8, 5};  
 finish[] = {2, 4, 6, 7, 9, 9};  
A person can perform at most four activities. The maximum set of activities that can be executed is {0, 1, 3, 4} [ These are indexes in start[] and finish[] ]

**Problem B**

Given an array of jobs where every job has a deadline and associated profit if the job is finished before the deadline. It is also given that every job takes single unit of time, so the minimum possible deadline for any job is 1. How to maximize total profit if only one job can be scheduled at a time.

**Sample Input/Output**

Input: Four Jobs with following deadlines and profits  
 JobID Deadline Profit  
 a 4 20   
 b 1 10  
 c 1 40   
 d 1 30  
Output: Following is maximum profit sequence of jobs  
 c, a

Input: Five Jobs with following deadlines and profits  
 JobID Deadline Profit  
 a 2 100  
 b 1 19  
 c 2 27  
 d 1 25  
 e 3 15  
Output: Following is maximum profit sequence of jobs  
 c, a, e

**Problem C**

Given arrival and departure times of all trains that reach a railway station, find the minimum number of platforms required for the railway station so that no train waits.

We are given two arrays which represent arrival and departure times of trains that stop

**Sample Input/Output**

Input: arr[] = {9:00, 9:40, 9:50, 11:00, 15:00, 18:00}  
 dep[] = {9:10, 12:00, 11:20, 11:30, 19:00, 20:00}  
Output: 3  
There are at-most three trains at a time (time between 11:00 to 11:20)

**Problem D**

Given two arrays a[] and b[] of equal length n. The task is to pair each element of array a to an element in array b, such that sum S of absolute differences of all the pairs is minimum.

Suppose, two elements a[i] and a[j] (i != j) of a are paired with elements b[p] and b[q] of b respectively, then p should not be equal to q.

**Sample Input/Output**

Input : a[] = {3, 2, 1}  
 b[] = {2, 1, 3}  
Output : 0  
  
Input : n = 4  
 a[] = {4, 1, 8, 7}  
 b[] = {2, 3, 6, 5}  
Output : 6