

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: 10 1 3 0 5 3 5 8 8 2 12 4 5 6 7 9 9 11 12 14 16	Output: 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: 7 2 1 3 2 2 2 1 2 3 8 6 2 5 3	Output: 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: 9 4 4 2 3 2 2 3 2 2	Output: yes 2
---	----------------------------

Week 11:

- I. Given a sequence of matrices, write an algorithm to find most efficient way to multiply these matrices together. To find the optimal solution, you need to find the order in which these matrices should be multiplied.

Input Format:

First line of input will take number of matrices n that you need to multiply.

For each line i in n, take two inputs which will represent dimensions aXb of matrix i.

Output Format:

Output will be the minimum number of operations that are required to multiply the list of matrices.

Sample I/O Problem I:

Input: 3 10 30 30 5 5 60	Output: 4500
---	------------------------

Solved Example: Consider a sequence of three matrices A of size 10X30, B of size 30X5, C of size 5X60. Then,

$$(AB)C = (10*30*5) + (10*5*60) = 4500 \text{ operations}$$

$$A(BC) = (30*5*60) + (10*30*60) = 27000 \text{ operations.}$$

Hence the output of the program must be 4500

- II. Given a set of available types of coins. Let suppose you have infinite supply of each type of coin. For a given value N, you have to Design an algorithm and implement it using a program to find number of ways in which these coins can be added to make sum value equals to N.

Input Format:

First line of input will take number of coins that are available.

Second line of input will take the value of each coin.

Third line of input will take the value N for which you need to find sum.

Output Format:

Output will be the number of ways.

Sample I/O Problem II:

Input: 4 2 5 6 3 10	Output: 5
-------------------------------------	---------------------

Solved Example: Let coin value set is $C = \{2, 3, 6, 5\}$ and the value $N = 10$. There are five solutions: $\{2, 2, 2, 2, 2\}$, $\{2, 2, 3, 3\}$, $\{2, 2, 6\}$, $\{2, 3, 5\}$ and $\{5, 5\}$. Hence the output is 5.

III. Given a set of elements, you have to partition the set into two subsets such that the sum of elements in both subsets is same. Design an algorithm and implement it using a program to solve this problem.

Input Format:

First line of input will take number of elements n present in the set.

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

Output Format:

Output will be 'yes' if two such subsets found otherwise print 'no'.

Sample I/O Problem III:

Input: 7 1 5 4 11 5 14 10	Output: yes
--	-----------------------

Solved Example: Let set is $S = \{1, 5, 4, 11, 5, 14, 10\}$. Sum of the elements = $1+5+4+11+5+14+10 = 50$. Now dividing the set into two halves such that sum of elements of both the subsets = $(50/2) = 25$. Therefore, subsets are $\{1, 5, 5, 14\}$ and $\{4, 11, 10\}$.