Competitive Programming Lab Questions

Problem Statement 1 – An automobile company manufactures both a two wheeler (TW) and a four wheeler (FW). A company manager wants to make the production of both types of vehicle according to the given data below:

- 1st data, Total number of vehicle (two-wheeler + four-wheeler) =v
- 2nd data, Total number of wheels = W

The task is to find how many two-wheelers as well as four-wheelers need to manufacture as per the given data.
Example :

Input :
200 -> Value of V
540 -> Value of W

Output :
TW =130 FW=70

Problem Statement 2 - Given an array of N integers arr [] where each element represents the maximum length of the jump that can be made forward from that element. This means if arr[i] = x, then we can jump any distance y such that $y \leq x$. Find the minimum number of jumps to reach the end of the array (starting from the first element). If an element is 0, then you cannot move through that element.

Note: Return -1 if you can't reach the end of the array.

Input: N = 11
arr [] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}

Output: 3

Explanation: First jump from 1st element to 2nd element with value 3. Now, from here we jump to 5th element with value 9, and from here we will jump to the last.

Problem Statement 3 - A doctor has a clinic where he serves his patients. The doctor's consultation fees are different for different groups of patients depending on their age. If the patient's age is below 17, fees is 200 INR. If the patient's age is between 17 and 40, fees is

400 INR. If patient's age is above 40, fees is 300 INR. Write a code to calculate earnings in a day for which one array/List of values representing age of patients visited on that day is passed as input.

Note:

- Age should not be zero or less than zero or above 120
- Doctor consults a maximum of 20 patients a day
- Enter age value (press Enter without a value to stop):

Example :
Input
20
30
40
50
2
3
14
Output
Total Income 2000 INR
Note: Input and Output Format should be same as given in the above example.
For any wrong input display INVALID INPUT
Output Format
Total Income 2100 INR

Problem Statement 4 - A party has been organised on cruise. The party is organised for a limited time(T). The number of guests entering (E[i]) and leaving (L[i]) the party at every hour is represented as elements of the array. The task is to find the maximum number of guests present on the cruise at any given instance within T hours.

Example 1:
Input :

- 5 -> Value of T
- [7,0,5,1,3] -> E [], Element of E [0] to E[N-1], where input each element is separated by new line
- [1,2,1,3,4] -> L [], Element of L [0] to L[N-1], while input each element is separate by new line.

Output :
8 -> Maximum number of guests on cruise at an instance.

Problem Statement 5 – Given an array of integers that may contain both positive and negative integers. Write a program to find all the pair of integers whose sum is equal to the desired sum.

Problem Statement 6 – Write a program for Post Order Traversal without Recursion.

Problem Statement 7 - Given an array **Arr[]** of **N** integers. Find the contiguous sub-array (containing at least one number) which has the maximum sum and return its sum.

Example -

Input:
N = 5
Arr[] = {1,2,3, -2,5 }
Output:
9
Explanation:
Max subarray sum is 9 of elements (1, 2, 3, -2, 5) which is a contiguous subarray.

Problem Statement 8 - Given an array **A** of size **N** which contains elements from **0** to **N-1**, you need to find all the elements occurring more than once in the given array. Return the answer in ascending order. If no such element is found, return list containing **[-1]**.

Note: The extra space is only for the array to be returned. Try and perform all operations within the provided array.

Example 1:

Input:
N = 4
A[] = {0,3,1,2}
Output:
-1
Explanation:
There is no repeating element in the array. Therefore, output is -1.

Example 2:

Input:
N = 5

A[] = {2,3,1,2,3}
Output:
2 3
Explanation:
2 and 3 occur more than once in the given array.

Problem Statement 9 – Write a program explaining Kadane's Algorithm.

Problem Statement 10 - Write a Program to reverse the Linked List. (Both Iterative and recursive).

Example –

Input:

LinkedList: 2->7->8->9->10

Output: 10 9 8 7 2

Explanation: After reversing the list,

elements are 10->9->8->7->2.

Problem Statement 11 - Given **head**, the head of a singly linked list, find if the linked list is circular or not. A linked list is called circular if it not NULL terminated and all nodes are connected in the form of a cycle. An empty linked list is considered as circular.

Note: The linked list does not contain any inner loop.

Input:

LinkedList: 1->2->3->4->5

(the first and last node is connected, i.e. 5 --> 1)

Output: 1

Problem Statement 12 - Given *weights* and *values* of **N** items, we need to put these items in a knapsack of capacity **W** to get the *maximum* total value in the knapsack.

Note: Unlike 0/1 knapsack, you are allowed to break the item.

Example –

Input:
N = 3, W = 50
values [] = {60,100,120}
weight [] = {10,20,30}
Output:
240.00
Explanation: Total maximum value of item we can have is 240.00 from the given capacity of sack.

Problem Statement 13 – Solve N-Queens Problem. The n-queens puzzle is the problem of placing n queens on an n x n chessboard such that no two queens attack each other.

Problem Statement 14 - Given two strings, find the length of longest subsequence present in both of them. Both the strings are in **uppercase** Latin alphabets.

Example –

Input:
A = 6, B = 6
str1 = ABCDGH
str2 = AEDFHR
Output: 3
Explanation: LCS for input strings "ABCDGH" and "AEDFHR" is "ADH" of length 3.

Problem Statement 15 - Given an integer **K** and a queue of integers, we need to reverse the order of the first **K** elements of the queue, leaving the other elements in the same relative order.

Only following standard operations are allowed on queue.

- enqueue(x) : Add an item x to rear of queue
- dequeue() : Remove an item from front of queue
- size() : Returns number of elements in queue.
- front() : Finds front item.
  Note: The above operations represent the general processing's. In-built functions of the respective languages can be used to solve the problem.

Example –

Input:
5 3
1 2 3 4 5

Output:
3 2 1 4 5

Explanation:
After reversing the given input from the 3rd position the resultant output will be 3 2 1 4 5.

Problem Statement 16 - Given a binary tree of size **N.** Your task is to complete the function **sumOfLongRootToLeafPath(),** that find the sum of all nodes on the longest path from root to leaf node.
If two or more paths compete for the longest path, then the path having maximum sum of nodes is being considered.

Example -

Input:

Output: 13

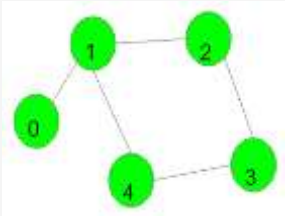
Explanation:

The highlighted nodes (**4, 2, 1, 6**) above are part of the longest root to leaf path having sum = (4 + 2 + 1 + 6) = 13

Problem Statement 17 - Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not. Graph is in the form of adjacency list where adj[i] contains all the nodes ith node is having edge with. Return 1 for yes (containing cycle) otherwise 0.

Example -

Input:
V = 5, E = 5
adj = {{1}, {0, 2, 4}, {1, 3}, {2, 4}, {1, 3}}
Output: 1
Explanation:



1->2->3->4->1 is a cycle.

Problem Statement 18 - There are given **N** ropes of different lengths, we need to connect these ropes into one rope. The cost to connect two ropes is equal to sum of their lengths. The task is to connect the ropes with minimum cost. Given **N** size array **arr[]** contains the lengths of the ropes.

Example –

Input:
n = 4
arr[] = {4, 3, 2, 6}
Output:
29
Explanation:
We can connect the ropes in following ways.
1) First connect ropes of lengths 2 and 3. Which makes the array {4, 5, 6}. Cost of this operation 2+3 = 5.
2) Now connect ropes of lengths 4 and 5. Which makes the array {9, 6}. Cost of this operation 4+5 = 9.
3) Finally connect the two ropes and all ropes have connected. Cost of this operation 9+6 =15
Total cost for connecting all ropes is 5 + 9 + 15 = 29. This is the optimized cost for connecting ropes.
Other ways of connecting ropes would always have same or more cost. For example, if we connect 4 and 6 first (we get three rope of 3, 2 and 10), then connect 10 and 3 (we get

two rope of 13 and 2). Finally, we connect 13 and 2. Total cost in this way is 10 + 13 + 15 = 38.

Problem Statement 19 - You have given a string. You need to remove all the duplicates from the string. The final output string should contain each character only once. The respective order of the characters inside the string should remain the same.

You can only traverse the string at once.

Example -

```
Input string: ababacd
Output string: abcd
```

Problem Statement 20 – Implement Priority Queue.