Data Structure and Practices
Assignment 3

Question 1 - (Hashing)

You are given an array A of integers of size N and a non-negative integer k,
Find all the indices i and j for which :
2*A[i] - A[j] = k and i!=j.

Expected time complexity - **O(N)**

-1_00_000 < A[i] <1_00_000
0 < N < 1_00_000
0< k < 1_00_000

Input Format:
N: number of integers
A: array containing the integers
K: the non-negative integer

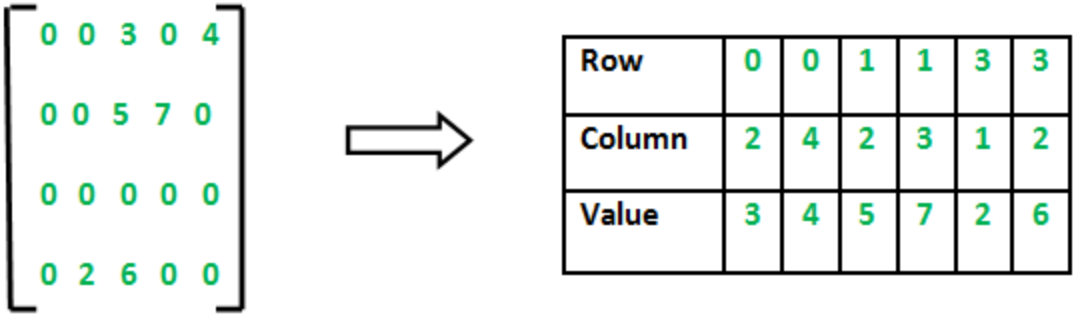
Output format:
The number of instances for which the condition is true.

Test case:
Input:
3
[2 , 6, 4]
2

Output:
2

Question 2: (Sparse Matrix)

Create an application which implements a sparse matrix in the format of triplet representation. Triplet representation of the sparse matrix is the way of representing each element of a 2-D matrix in the form of a triplet (row, column, element) stored as an array.



The application should provide specified options at runtime given below:

1- Initialize_sparse_matrix:

Ask the rows and columns as input. Then initialize a 2-D matrix of given size with random values.

2- Print the sparse matrix:

Although the matrix is saved in triplet form but on display it should be printed in sparse 2-D matrix form. Don't use any extra space for this function. Hint: sorting.

3- Delete an element:

Take the row and column index and make it zero. If already zero then do nothing.

4- add an element:

Take the row and column index and the value of the element and add it in the sparse matrix.

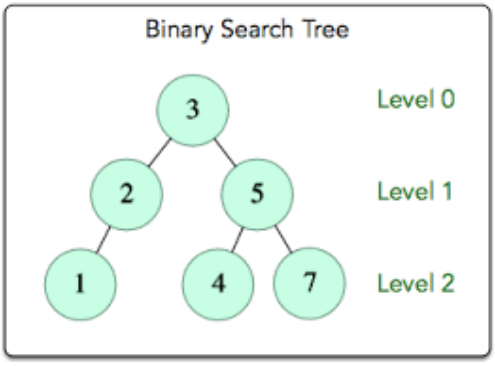
Question 3. [BST]

Implement a BST by making nodes (using Classes, Struct, etc). Make functions of Insert, Search, Delete, and Print the BST. After every insertion/ Deletion print the BST.

Printing of BST should be done using level order.

The search function should return the height (root height ==0) and the position of that element from left to right. If not found return -1 for both.

Example :



For this BST

The print function should return : {3}, {2,5}, {1,,4,7}

Search(4) should return : Level =2 , Position =2

Question 4: Implement the BST using an array. Make functions of Insert, Search, Delete, and Print the BST. After every insertion/ Deletion print the BST.

Printing of BST should be done using level order.

Given an Input value K, the search function should return if there are two values A and B in the BST such that

A+B=K

If found print A and B else return -1.