

Week 5:

I. Given an unsorted array of alphabets containing duplicate elements. Design an algorithm and implement it using a program to find which alphabet has maximum number of occurrences and

print it. (Time Complexity = $O(n)$) (Hint: Use counting sort)

Input Format:

The first line contains number of test cases, T.
For each test case, there will be two input lines.
First line contains n (the size of array).
Second line contains space-separated integers describing array.

Output:

The output will have T number of lines.
For each test case, output will be the array element which has maximum occurrences and its total number of occurrences.
If no duplicates are present (i.e. all the elements occur only once), output should be “No Duplicates Present”.

Sample I/O Problem I:

Input: 3 10 a e d w a d q a f p 15 r k p g v y u m q a d j c z e 20 g t l l t c w a w g l c w d s a a v c l	Output: a - 3 No Duplicates Present l - 4
-----------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------

II. Given an unsorted array of integers, design an algorithm and implement it using a program to find whether two elements exist such that their sum is equal to the given key element. (Time Complexity = $O(n \log n)$)

Input Format:

The first line contains number of test cases, T.
For each test case, there will be two input lines.
First line contains n (the size of array).
Second line contains space-separated integers describing array.
Third line contains key

Output Format:

The output will have T number of lines.
For each test case, output will be the elements arr[i] and arr[j] such that arr[i]+arr[j] = key if exist otherwise print 'No Such Elements Exist'.

Sample I/O Problem II:

Input: 2 10 64 28 97 40 12 72 84 24 38 10 50 15 56 10 72 91 29 3 41 45 61 20 11 39 9 12 94 302	Output: 10 40 No Such Element Exist
----------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------

III. You have been given two sorted integer arrays of size m and n . Design an algorithm and implement it using a program to find list of elements which are common to both. (Time Complexity = $O(m+n)$)

Input Format:

First line contains m (the size of first array).

Second line contains m space-separated integers describing first array.

Third line contains n (the size of second array).

Fourth line contains n space-separated integers describing second array.

Output Format:

Output will be the list of elements which are common to both.

Sample I/O Problem III:

Input: 7 34 76 10 39 85 10 55 12 30 55 34 72 10 34 10 89 11 30 69 51	Output: 10 10 34 55
-----------------------------------------------------------------------------------------	-------------------------------