**Arrays:**

1. Write an efficient program to find the sum of contiguous subarray within a one-dimensional array of numbers which has the largest sum.
2. You are given a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in list. One of the integers is missing in the list. Write an efficient code to find the missing integer.
3. Given an unsorted array of nonnegative integers, find a continous subarray which adds to a given number.

Examples :

Input: arr[] = {1, 4, 20, 3, 10, 5}, sum = 33

Ouptut: Sum found between indexes 2 and 4

Input: arr[] = {1, 4, 0, 0, 3, 10, 5}, sum = 7

Ouptut: Sum found between indexes 1 and 4

Input: arr[] = {1, 4}, sum = 0

Output: No subarray found

1. Given an unsorted array of integers, find a subarray which adds to a given number. If there are more than one subarrays with sum as the given number, print any of them.

Examples:

Input: arr[] = {1, 4, 20, 3, 10, 5}, sum = 33

Ouptut: Sum found between indexes 2 and 4

Input: arr[] = {10, 2, -2, -20, 10}, sum = -10

Ouptut: Sum found between indexes 0 to 3

Input: arr[] = {-10, 0, 2, -2, -20, 10}, sum = 20

Ouptut: No subarray with given sum exists

1. Write a program to sort an array of 0's,1's and 2's in ascending order.
2. Equilibrium index of an array is an index such that the sum of elements at lower indexes is equal to the sum of elements at higher indexes. For example, in an array A:

Example :

Input : A[] = {-7, 1, 5, 2, -4, 3, 0}

Output : 3

3 is an equilibrium index, because:

A[0] + A[1] + A[2]  =  A[4] + A[5] + A[6]

Write a function int equilibrium(int[] arr, int n); that given a sequence arr[] of size n, returns an equilibrium index (if any) or -1 if no equilibrium indexes exist.

1. Write a program to print all the LEADERS in the array. An element is leader if it is greater than all the elements to its right side. And the rightmost element is always a leader. For example int the array {16, 17, 4, 3, 5, 2}, leaders are 17, 5 and 2.
2. Given an array and a number k where k is smaller than size of array, we need to find the k’th smallest element in the given array. It is given that ll array elements are distinct.

Examples:

Input: arr[] = {7, 10, 4, 3, 20, 15}

       k = 3

Output: 7

Input: arr[] = {7, 10, 4, 3, 20, 15}

       k = 4

Output: 10

1. Given a 2D array, print it in spiral form. See the following examples.

Examples:

Input:

        1    2   3   4

        5    6   7   8

        9   10  11  12

        13  14  15  16

Output:

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Input:

        1   2   3   4  5   6

        7   8   9  10  11  12

        13  14  15 16  17  18

Output:

1 2 3 4 5 6 12 18 17 16 15 14 13 7 8 9 10 11

1. Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

Examples:

Input:  arr[] = {2, 5, 2, 8, 5, 6, 8, 8}

Output: arr[] = {8, 8, 8, 2, 2, 5, 5, 6}

Input: arr[] = {2, 5, 2, 6, -1, 9999999, 5, 8, 8, 8}

Output: arr[] = {8, 8, 8, 2, 2, 5, 5, 6, -1, 9999999}

**Stacks and Queues:**

1. Given an array, print the Next Greater Element (NGE) for every element. The Next greater Element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1.

Examples:  
a) For any array, rightmost element always has next greater element as -1.  
b) For an array which is sorted in decreasing order, all elements have next greater element as -1.  
c) For the input array [4, 5, 2, 25}, the next greater elements for each element are as follows.

Element       NGE

   4      -->   5

   5      -->   25

   2      -->   25

   25     -->   -1

d) For the input array [13, 7, 6, 12}, the next greater elements for each element are as follows.

  Element        NGE

   13      -->    -1

   7       -->     12

   6       -->     12

   12     -->     -1

1. Implement a Queue using 2 stacks**s1** and**s2** .
2. Implement a Stack using 2 queue**q1** and**q2** .
3. Implement a Stack in which you can get min element in O(1) time and O(1) space.
4. The task is to design and implement methods of an **LRU cache**. The class has two methods get and set which are defined as follows.  
   get(x)   : Gets the value of the key x if the key exists in the cache otherwise returns -1  
   set(x,y) : inserts the value if the key x is not already present. If the cache reaches its capacity it should invalidate the least recently used item before inserting the new item.  
   In the constructor of the class the size of the cache should be initialized.
5. Given an input stream of n characters consisting only of small case alphabets the task is to find the first non repeating character each time a character is inserted to the stream. If no non repeating element is found print -1.  
     
   Example

Flow in stream : a, a, b, c

a goes to stream : 1st non repeating element a (a)

a goes to stream : no non repeating element -1 (a, a)

b goes to stream : 1st non repeating element is b (a, a, b)

c goes to stream : 1st non repeating element is b (a, a, b, c)

**Strings:**

1. Given a string S, find the longest palindromic substring in S.

**Input:** "babad"

**Output:** "bab"

**Note:** "aba" is also a valid answer.

1. Given a string, recursively remove adjacent duplicate characters from string. The output string should not have any adjacent duplicates. See following examples.

Input:  azxxzy

Output: ay

First "azxxzy" is reduced to "azzy". The string "azzy" contains duplicates, so it is further reduced to "ay".

Input: caaabbbaacdddd

Output: Empty String

Input: acaaabbbacdddd

Output: acac

1. Given two strings, the task is to find if a string ('a') can be obtained by rotating another string ('b') by two places.  
   Examples:

Input : a = "amazon"         b = "azonam"  // rotated anti-clockwise

Output : True

Input : a = "amazon"        b = "onamaz"  // rotated clockwise

Output : True

1. Given an string in roman no format (s)  your task is to convert it to integer .

**Example:  
Input**  
2  
V  
III   
**Output**  
5  
3

1. Given two strings ‘X’ and ‘Y’, find the length of the longest common substring.

Input : X = "abcdxyz", y = "xyzabcd"

Output : 4

The longest common substring is "abcd" and is oflength 4.

Input : X = "zxabcdezy", y = "yzabcdezx"

Output : 6

The longest common substring is "abcdez" and is oflength 6.

1. Given a string, find length of the longest substring with all distinct characters.  For example, for input "abca", the output is 3 as "abc" is the longest substring with all distinct characters.
2. Your task  is to implement the function **atoi**. The function takes a string(str) as argument and converts it to an integer and returns it.
3. Your task  is to implement the function **strstr**. The function takes two strings as arguments(s,x) and  locates the occurrence of the string x in the string s. The function returns and integer denoting  the first occurrence of the string x .
4. Given a array of **N**strings, find the longest common prefix among all strings present in the array.

Input: apple ape April

Output: ap

1. Design a URL shortner like bit.ly

**LinkedList:**

1. Given a singly linked list, find middle of the linked list. For example, if given linked list is 1->2->3->4->5 then output should be 3.

If there are even nodes, then there would be two middle nodes, we need to print second middle element. For example, if given linked list is 1->2->3->4->5->6 then output should be 4.

1. Given a singly linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer smaller than or equal to length of the linked list. For example, if the given linked list is 10->20->30->40->50->60 and k is 4, the list should be modified to 50->60->10->20->30->40.
2. Given a linked list, write a function to reverse every k nodes (where k is an input to the function).If a linked list is given as 1->2->3->4->5->6->7->8->NULL and k = 3 then output will be 3->2->1->6->5->4->8->7->NULL.
3. Given a linked list, check if the the linked list has a loop. Linked list can contain self loop.
4. Given a linked list, the task is to find the n'th node from the end.
5. Given two linked lists sorted in ascending order. Merge them in-place and return head of the merged list.   For example, if first list is 10->20->30 and second list is 15->17, then the result list should be 10->15->17->20->30.

It is strongly recommended to do merging in-place using O(1) extra space.

1. Given a **L**inked **L**ist where every node represents a linked list and contains two pointers of its type:  
   (i) a**next**pointer to the next node  
   (ii) a**bottom** pointer to a linked list where this node is head.  
     
   You have to**flatten** the linked list to a **single linked list** which is

**For Ex:**Shown below is a given linked list

           5 -> 10 -> 19 -> 28       |    |     |     |       V    V     V     V       7    20    22    35       |          |     |       V          V     V       8          50    40       |                |       V                V       30               45

and after flattening it, the sorted linked list looks like:

 5->7->8->10->19->20->22->28->30->35->40->45->50  
  
The  node structure has **3** fields as mentioned. A **next pointer**, a **bottom** pointerand a **data** part.

1. Given a singly linked list, write a function to swap elements pairwise. For example, if the linked list is 1->2->3->4->5 then the function should change it to 2->1->4->3->5, and if the linked list is 1->2->3->4->5->6 then the function should change it to 2->1->4->3->6->5.
2. Given two numbers represented by two lists, write a function that returns sum list. The sum list is list representation of addition of two input numbers.

Suppose you have two linked list 5->4 ( 4 5 )and 5->4->3( 3 4 5) you have to return  a resultant linked list 0->9->3 (3 9 0).

10. Given a singly linked list of integers, Your task is to complete the function **isPalindrome** that returns true if the given list is palindrome, else returns false.