# List of Programming Questions

## => 1. Implement a LRU cache //LRUCache.java

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.

Input:

The first line of input contain an integer T denoting the no of test cases. Then T test case follow. Each test case contains 3 lines. The first line of input contains an integer N denoting the capacity of the cache and then in the next line is an integer Q denoting the no of queries Then Q queries follow. A Query can be of two types

1. SET x y : sets the value of the key x with value y

2. GET x : gets the key of x if present else returns -1.

Output:

For each test case in a new line output will be space separated values of the query.

Constraints:

1<=T<=100

1<=N<=10

1<=Q<=100

Example(To be used only for expected output):

Input

2

2

2

SET 1 2 GET 1

2

7

SET 1 2 SET 2 3 SET 1 5 SET 4 5 SET 6 7 GET 4 GET 1

Output

2

5 -1

## => 2. Numbers with same first and last digit //NumbersWithSameFirstAndLastDigit.java

Given an range of number's starting with integer L and ending at R, the task is to count the numbers which have same first and last digits.

Input:

The first line of the input contains an integer T, denoting the number of test cases. The T test case follow. The only line of the each test case contains two space seperated integer's L and R.

Output:

For each test case output a single line containing a single integer denoting the count of the required number's.

Constraints:

1<=T<=104

1<=L, R<=10^8

Example:

Input:

2

7 68

5 40

Output:

9

8

## => 3. Sort Array with respect to another given array //SortArrayWRTAnotherArray.java

Given two arrays A1[] and A2[], sort A1 in such a way that the relative order among the elements will be same as those are in A2. For the elements not present in A2, append them at last in sorted order.

Input: A1[] = {2, 1, 2, 5, 7, 1, 9, 3, 6, 8, 8}

A2[] = {2, 1, 8, 3}

Output: A1[] = {2, 2, 1, 1, 8, 8, 3, 5, 6, 7, 9}

The code should handle all cases like number of elements in A2[] may be more or less compared to A1[]. A2[] may have some elements which may not be there in A1[] and vice versa is also possible.

## => 4. Reverse a Linked List (Singly Linked List) //LinkedListReversal.java

Two approaches: 1. Use a stack

2. Just change the pointers (Iterative approach) //In comments

3. Recursive Approach //In same program

## => 5. Reverse a String //StringReversal.java

Reverse and print the string

## => 6. Whether s1 is a permutation of s2 //CheckPermutation.java

Check whether s1 is a permutation of s2

## => 7. Whether s2 is a rotation of s1 //CheckRotation.java

Check whether s2 is a rotation of s1.

Eg s1=waterbottle s2=erbottlewat

## => 8. Total number of negative numbers in sorted array //TotalNegNosInSortedArray.java

Find the total number of negative numbers in a row wise and column wise sorted array

## => 9. Maximum Sum of a Subarray //MaxSumSubarray.java

Maximum sum of a subarray (Kadane’s algo O(n) time or else O(n2) time)

## => 10. Kth to Last element in a linked list //KTHToLastElement.java

Find the Kth to last element in a linked list.

## => 11. Remove duplicates from unsorted linked list //RemoveDuplicatesLinkedList.java

Remove duplicated from unsorted linked list

With the use of a buffer (HashSet)

Without a use of a buffer

## => 12. Derivative of polynomial //Polynomial.java

## => 13. Replace spaces with %20 in a string

Replace all spaces in a string with %20.