1. Minimum number of deletions to make a string palindrome

Given a string of size ‘n’. The task is to remove or delete the minimum number of characters from the string so that the resultant string is a palindrome.

Note: The order of characters should be maintained.

Examples :

Input : aebcbda

Output : 2

Remove characters 'e' and 'd'

Resultant string will be 'abcba'

which is a palindromic string

2. **Two strings str1 and str2 are called isomorphic if there is a one-to-one mapping possible for every character of str1 to every character of str2. And all occurrences of every character in ‘str1’ map to the same character in ‘str2’.**

Examples:

Input: str1 = “aab”, str2 = “xxy”
Output: True
Explanation: ‘a’ is mapped to ‘x’ and ‘b’ is mapped to ‘y’.

Input: str1 = “aab”, str2 = “xyz”
Output: False
Explanation: One occurrence of ‘a’ in str1 has ‘x’ in str2 and other occurrence of ‘a’ has ‘y’.

3. Given a string, find the count of distinct subsequences of it.
Examples:

Input: str = “gfg”
Output: 7
Explanation: The seven distinct subsequences are “”, “g”, “f”, “gf”, “fg”, “gg” and “gfg”

Input: str = “ggg”
Output: 4
Explanation: The four distinct subsequences are “”, “g”, “gg” and “ggg”

4.**Merge a linked list into another linked list at alternate positions**

Given two linked lists, insert nodes of second list into first list at alternate positions of first list.

For example, if first list is 5->7->17->13->11 and second is 12->10->2->4->6, the first list should become 5->12->7->10->17->2->13->4->11->6 and second list should become empty. The nodes of second list should only be inserted when there are positions available. For example, if the first list is 1->2->3 and second list is 4->5->6->7->8, then first list should become 1->4->2->5->3->6 and second list to 7->8.

5.Given a singly linked list, The task is to rotate the linked list counter-clockwise by k nodes.

Examples:

Input: linked list = 10->20->30->40->50->60, k = 4
Output: 50->60->10->20->30->40.
Explanation: k is smaller than the count of nodes in a linked list so (k+1)th node i.e. **50** becomes the head node and 60’s next points to 10

Input: linked list = 30->40->50->60, k = 2
Output: 50->60->30->40.

6. Given a singly linked list, write a function to swap elements pairwise.

Input : 1->2->3->4->5->6->NULL
Output : 2->1->4->3->6->5->NULL

Input : 1->2->3->4->5->NULL
Output : 2->1->4->3->5->NULL

Input : 1->NULL
Output : 1->NULL

7. Given a matrix, check whether it's Magic Square or not. A Magic Square is a n x n matrix of the distinct elements from 1 to n2 where the sum of any row, column, or diagonal is always equal to the same number.

Examples:

Input : n = 3
  2   7   6
  9   5   1
  4   3   8

Output : Magic matrix
Explanation:In matrix sum of each
row and each column and diagonals sum is
same = 15.

Input : n = 3
 1 2 2
 2 2 1
 2 1 2
Output : Not a Magic Matrix
Explanation:In matrix sum of each
row and each column and diagonals sum is
not same.

8. Count all sorted rows in a matrix

Given a matrix of m*n size, the task is to count all the rows in a matrix that are sorted either in strictly increasing order or in strictly decreasing order?

Examples :

```
Input : m = 4,  n = 5
        mat[m][n] = 1 2 3 4 5
                    4 3 1 2 6
                    8 7 6 5 4
                    5 7 8 9 10
Output: 2
```

9. Find median in row wise sorted matrix

We are given a row-wise sorted matrix of size r*c, we need to find the median of the matrix given. It is assumed that r*c is always odd.

Examples:

Input:
1 3 5
2 6 9
3 6 9
Output:
Median is 5
If we put all the values in a sorted array A[] = 1 2 3 3 5 6 6 9 9)

Input:
1 3 4
2 5 6
7 8 9
Output:
Median is 5

10 Length of the longest valid substring

Given a string consisting of opening and closing parenthesis, find the length of the longest valid parenthesis substring.

Examples:

Input: ((()
Output : 2
Explanation : ()
Input:)()())
Output : 4
Explanation: ()()
Input: ()(()))))
Output: 6
Explanation: ()(())

11. Next Greater Frequency Element

Given an array, for each element find the value of the nearest element to the right which is having a frequency greater than that of the current element. If there does not exist an answer for a position, then make the value '-1'.

Examples:

Input : a[] = [1, 1, 2, 3, 4, 2, 1]
Output : [-1, -1, 1, 2, 2, 1, -1]

12. Check if two expressions with brackets are same

Given two expressions in the form of strings. The task is to compare them and check if they are similar. Expressions consist of lowercase alphabets, ‘+’, ‘-‘ and ‘( )’.

Examples:

```
Input  : exp1 = "-(a+b+c)"
         exp2 = "-a-b-c"
Output : Yes

Input  : exp1 = "-(c+b+a)"
         exp2 = "-c-b-a"
Output : Yes

Input  : exp1 = "a-b-(c-d)"
         exp2 = "a-b-c-d"
Output : No
```

13. Reversing a queue using recursion

Given a queue, write a recursive function to reverse it.

Standard operations allowed :

enqueue(x) : Add an item x to rear of queue.
dequeue() : Remove an item from front of queue.
empty() : Checks if a queue is empty or not.

Examples :

```
Input : Q = [5, 24, 9, 6, 8, 4, 1, 8, 3, 6]
Output : Q = [6, 3, 8, 1, 4, 8, 6, 9, 24, 5]
```

Explanation : Output queue is the reverse of the input queue.

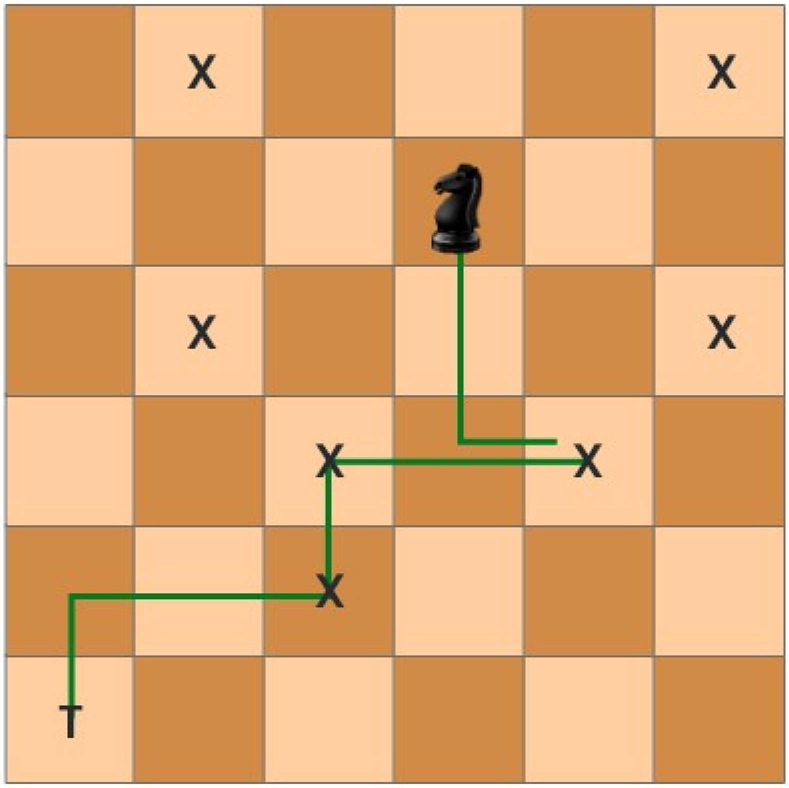
Input : Q = [8, 7, 2, 5, 1]
Output : Q = [1, 5, 2, 7, 8]

14. **Minimum steps to reach target by a Knight**

Given a square chessboard of **N x N** size, the position of the Knight and the position of a target are given. We need to find out the minimum steps a Knight will take to reach the target position.

Examples:

Input:



Knight

knightPosition: (1, 3) , targetPosition: (5, 0)

Output: 3

Explanation: In above diagram Knight takes 3 step to reach
from (1, 3) to (5, 0)
(1, 3) -> (3, 4) -> (4, 2) -> (5, 0)

15. Geek in a Maze

Geek is in a maze of size N * M. Each cell in the maze is made of either '.' or '#'. An empty cell is represented by '.' and an obstacle is represented by '#'. The task is to find out how many different empty cells he can pass through If Geek starts at cell (R, C) and avoids the obstacles and he can move in any of the four directions but he can move up at most U times and he can move down at most D times.

Examples:

Input: N = 3, M = 3,
R = 1, C = 0
U = 1, D = 1
mat = {{. . .}, {. # .}, {# . .}}
Output: 5
Explanation: Geek can reach
(1, 0), (0, 0), (0, 1), (0, 2), (1, 2)

Input: N = 3, M = 4, R = 1, C = 0, U = 1, D = 2
mat = {{. . .}, {. # .}, {. . .}, {# . .}}
Output: 10
Explanation: Geek can reach all the
cells except for the obstacles

16. **Union and Intersection of two sorted arrays**

Given two sorted arrays, find their union and intersection.

Example:

Input: arr1[] = {1, 3, 4, 5, 7}
arr2[] = {2, 3, 5, 6}
Output: Union : {1, 2, 3, 4, 5, 6, 7}
Intersection : {3, 5}

Input: arr1[] = {2, 5, 6}
arr2[] = {4, 6, 8, 10}
Output: Union : {2, 4, 5, 6, 8, 10}
Intersection : {6}

17 **Find the first repeating element in an array of integers**

Given an array of integers **arr[]**, The task is to find the index of first repeating element in it i.e. the element that occurs more than once and whose index of the first occurrence is the smallest.

Examples:

Input: arr[] = {10, 5, 3, 4, 3, 5, 6}

Output: 5

Explanation: 5 is the first element that repeats

Input: arr[] = {6, 10, 5, 4, 9, 120, 4, 6, 10}

Output: 6

Explanation: 6 is the first element that repeats

18 . **Rearrange array in alternating positive & negative items with O(1) extra space**

Given an array having **positive** and **negative numbers**, our task is to arrange them in an **alternate** fashion such that every positive number is followed by a negative number and vice-versa maintaining the **order of appearance**. The number of positive and negative numbers need not to be equal. If there are more positive numbers then they have to appear at the end of the array , same condition for negative numbers also .

Examples:

Input: arr[] = {1, 2, 3, -4, -1, 4}

Output: arr[] = {-4, 1, -1, 2, 3, 4}

Input: arr[] = {-5, -2, 5, 2, 4, 7, 1, 8, 0, -8}

Output: arr[] = {-5, 5, -2, 2, -8, 4, 7, 1, 8, 0}