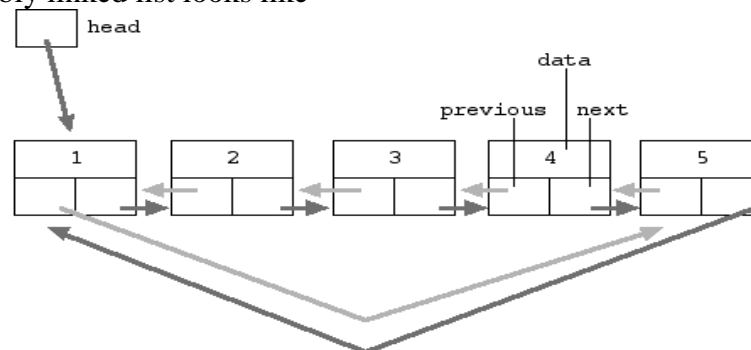


BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI HYDERABAD CAMPUS,
Data Structures and Algorithms
CS F211
Homework Assignment – 7

1.
 - a. Write a program to modify a singly linked list such that all even numbers appear before all the odd numbers in the modified linked list
For example: if the linked list is 1->2->3->4->5->6->NULL, then the resultant list should be 2->4->6->1->3->5->NULL
 - b. Given a singly linked list (create a linked list by adding element at the beginning or ending), write a function to swap elements pairwise. **For example:** if the linked list is 1->2->3->4->5->6->7->NULL, then the function should change it to 2->1->4->3->6->5->7->NULL.
2. Write a program to reverse 'M' elements of a singly linked list starting from the given location 'L' by providing the menu driven options such as,
 - a. Insert an element at the beginning
 - b. Insert an element at the end
 - c. Reverse sub part of the list (input M and L are required)
 - d. Print the elements of the list**Note:** You need to handle all exceptions (**Example:** Printing the elements of the empty list, reversing the sub part of the list where input M and L are not within the range of your current list, etc.).
3. Write a program to implement circular doubly linked list by providing the menu driven options such as,
 - a. Insert an element at the beginning
 - b. Insert an element at the end
 - c. Insert an element after an index (input i - index)
 - d. Print the elements of the list

A circular doubly linked list looks like



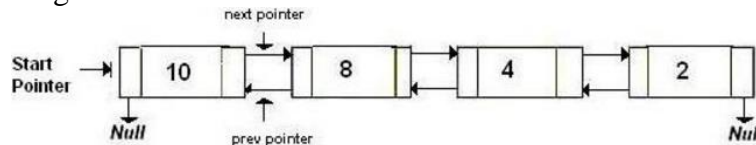
Note: You need to handle all exceptions (**Example:** Printing the elements of the empty list, inserting an element after an index which is greater than the no. of

elements in the list, etc.).

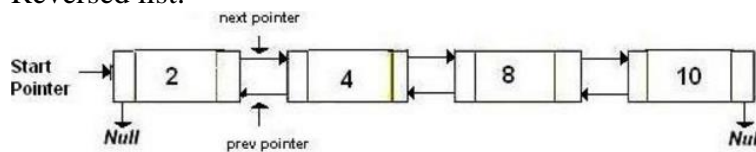
4. Create a doubly linked list and reverse the list by providing the menu driven options such as,
 - a. Insert an element at the beginning
 - b. Insert an element at the end
 - c. Reverse the elements of the list (1. Iterative method, 2. Recursive method)
 - d. Print the elements of the list

Example:

Original list:



Reversed list:



5. Write a program to check whether a singly linked list is palindrome or not. (**Note:** For creating the list make function InsertAtBeg to insert at beginning which will very easily create a list).
6. You all know the problem of parenthesization where a sequence of (or) is balanced or not. You are given a parenthesis sequence of length N consisting of (and). It may be valid or may be invalid (i.e. not balanced). But you can select any of it and flip it. A flip changes (to) or) to (. You have to find the minimum number of flips required to make the sequence balanced. For example : ((() needs 1 change to make it to (() and)(needs 2 changes to make it to (). You will have even number of characters in your string. You have to solve it using some data structure.

Sample Input :

((()))(

Sample Output:

2

7. After Test-I exams, you all went for a movie on Tuesday night. There was a very long queue for tickets at the Cinema Hall. N people are waiting in a queue for tickets. One of the guy Ravi Teja thought of a puzzle by seeing the long queue. Puzzle is that two person X and Y standing in the line can see each other if no person between them is **strictly taller** than X or Y. How many pairs of person will be able to see each other? Given the heights of all the N person in order, help Ravi Teja to solve this puzzle. Solve it in O(n) time [Hint : Use some data structure]

Input

First line contains an integer N

Second Line contains N integers representing the height of people in the order in which they are standing in a queue.

Output:

Print the number of pairs of persons that can see each other.

Sample Input:

7
1 1 4 3 5 2 6

Sample Output:

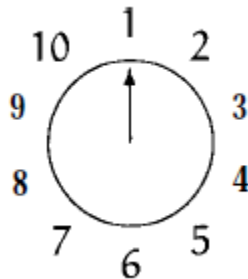
9

Explanation: 9 pairs are there who can see each other { (1,1), (1,4), (1,4), (4, 3), (4,5),(3,5),(5,2),(5,6),(2,6) }

8. People are standing in a circle waiting to be executed. Counting begins at a specified point in the circle and proceeds around the circle in a specified direction. After a specified number of people are skipped, the next person is executed. The procedure is repeated with the remaining people, starting with the next person, going in the same direction and skipping the same number of people, until only one person remains, and is freed. Given the total number of person's n and a number k which indicates that k-1 persons are skipped and kth person is executed in circle. Find which person will be given freedom.

You have to solve it using circular linked-list data structure.

Example: n = 10 and k = 2



Elimination order is 2, 4, 6, 8, 10, 3, 7, 1, 9 so 5 survives.

Input

N K

Output:

Print the person who is freed.

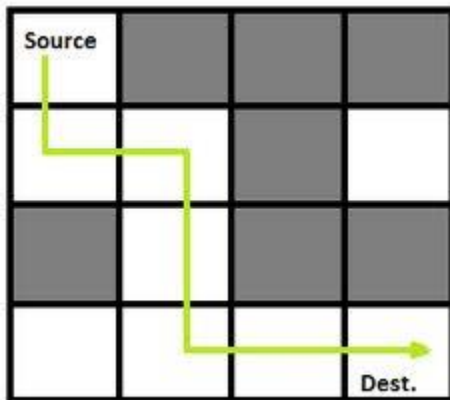
9. Write a program to convert a given correct infix expression into postfix expression, print the postfix expression, evaluate the postfix using stack and print the result.

You can assume that only the following operators are allowed:

+ - * / ()

10. You have a NxN maze in the form of a matrix. This maze will have some cells as

open and some cells will be closed (obstacles).



A Rat can move one cell at a time in one of the four directions with following priority:

First it tries to move to one cell right, if it can't move right because of obstacle or matrix bound, it tries to go Down, if it can't move Down also, it tries to move Left, lastly if it can't move in any of the other directions it tries to move Up. The rat tries to exhaust all these possibilities before backtracking to a path it has come from.

So the priority of directional movement is:

Right > Down > Left > Up

Initially, the Rat is at the top left position *i.e.* at matrix[0][0]

There's a way out at the bottom right which the Rat has to reach *i.e.* position matrix[N-1][N-1].

The rat starts to move one cell at a time from the beginning and its goal is to reach the bottom right position. If the rat finds a dead-end while trying to move and it has exhausted all its movement possibilities, it tries to backtrack upto the last position where it has some options open to move according to its priority order.

Write a program that uses your own stack for two different 10x10 matrices given below that will print the complete movement of the rat in reaching the exit at bottom right, or will print "No path exists to exit" and show its movements in the matrix after trying all the movements. If there is a path, the program should print how many times the rat has moved in a cell during the journey. In the matrix below, elements with 1 show path open and elements 0 mean it is closed.

The output should also be in the form of a matrix that shows all the paths the rat has tried and the final taken. If a cell is in the path taken by the rat, you can increment it by one for

Note: Do NOT use recursive calls in your program. Create and use explicit stack(s). Apply backtracking method by using the stack(s).

The matrices are:

```

1 1 1 1 1 1 1 0 0 0
1 0 1 1 0 0 0 0 0 0
1 1 0 0 1 1 0 0 0 0
1 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 0

```

1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 0 0 0
0 0 0 0 1 0 0 1 1 1
0 0 0 0 1 1 1 1 0 1
0 0 0 1 1 1 1 1 0 1

AND

1 1 1 1 1 1 1 0 0 0
1 0 1 1 0 0 0 0 0 0
1 1 0 0 1 1 0 0 0 0
1 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 0
1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 0 0 0
0 0 0 0 1 0 0 0 1 1
0 0 0 0 1 1 1 1 0 1
0 0 0 1 1 1 1 1 0 1

-----XXXXXXXXXXXXXXXXXXXXXXXXXXXX-----