Data Structures (CS -102) Fall 2015 **Section A, B**
National University of Computer and Emerging Sciences, Lahore

Assignment # 3

Recursion

Submission Dead Line: Monday 12/10/2015

LATE SUBMISSION WILL NOT BE ACCEPTED

Question 1: (Link Lists)

You have implemented Link List in Assignment #2, Now Add three recursive functions for searching, and printing in both (Forward & Reverse) order.

Question 2: (Recursive Algorithms)

Implement recursive solutions for problems given below. Identify recurrence relations for T(n) with both base and general case and solve to find time complexity Big-oh (n).

1. **Binary Search**.
   a. Compare the search element with the middle element of the array
   b. If not equal, then apply binary search to half of the array (if not empty) where the search element would be.

Input: integer array, key and size, return true if found element

```
bool BinarySearch(int arr[], int size, int key);
```

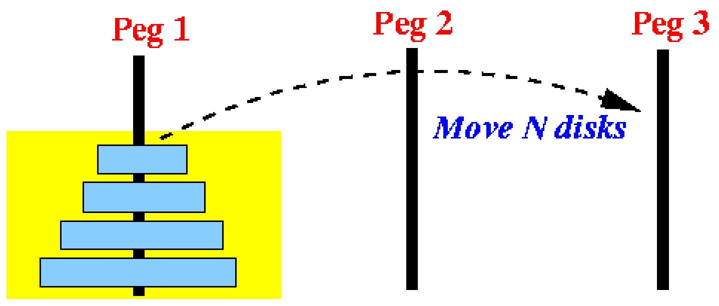
2. Find if a string is **Palindrome** or not.

A palindrome is a word or sentence that reads the same in both forward and backward directions for example, **MADAM**, **CIVIC, LEVEL, REFER, AAAA**

This function returns true, if string is palindrome. Input: A String of variable size

```
bool isPalindrome(char * str);
```

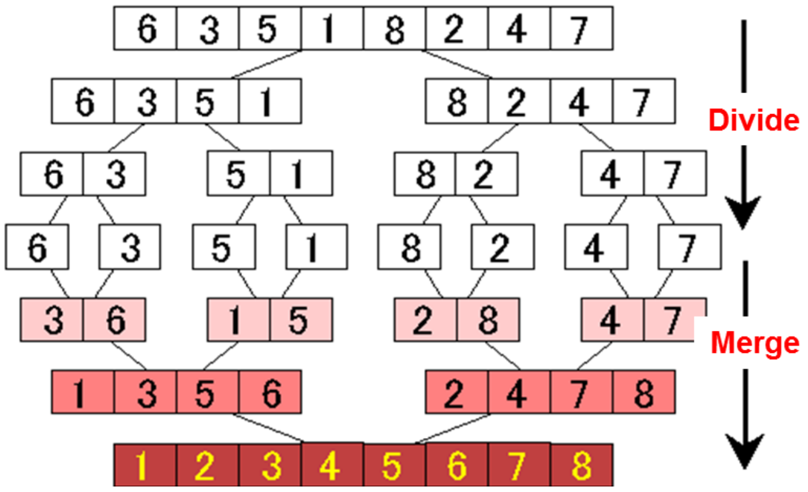
3. **Tower of Hanoi** Problem
   a. Only one disc could be moved at a time
   b. A larger disc must never be stacked above a smaller one
   c. One and only one extra needle could be used for intermediate storage of discs



Input: Source Tower, Destination tower, Num = Number of Disks to move

```
void hanoi(int from, int to, int num)
```

4. **Merge Sort**.
   In computer science, merge sort is an O(n log n) comparison-based sorting algorithm.
   a. Divide the unsorted list into n sublists, each containing 1 element (a list of 1 element is considered sorted).
   b. Repeatedly merge sublists to produce new sorted sublists until there is only 1 sublist remaining. This will be the sorted list.



Input: unsorted integer array, starting and ending indexes

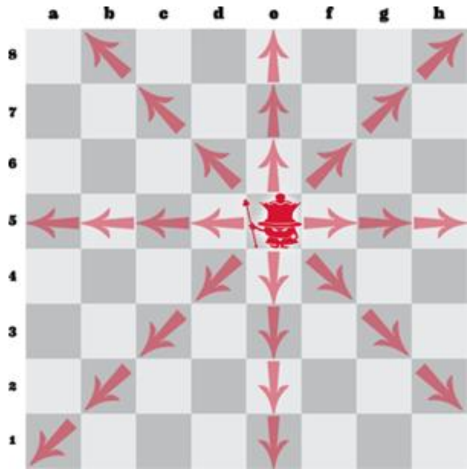
```
void MergeSort(int arr[], int low, int high);
```

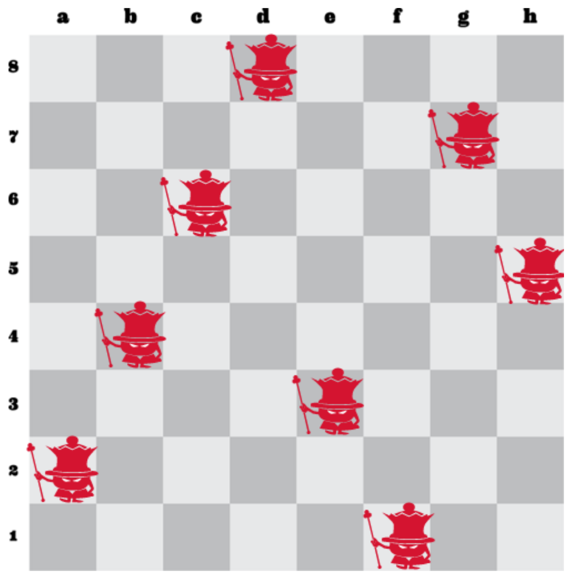
5. **Eight Queens** Problem.

   In chess the queen is the most powerful piece on the board. She can move up and down, side to side, and diagonally, as many squares as she wants. The eight Queens puzzle is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other. Thus, a solution requires that no two queens share the same row, column, or diagonal.

Input: integer array to store row numbers of queens as solution, rows and cols.

```
bool queens(int t[], int row, int col)
```





```
int main() {
     int SIZE = 8;
     int t[SIZE]={0};
     for (int i= 0; i <SIZE; i++){
          t[0] = i;
//place all queens in first row, but different columns
    if (queens(t, 1,0))
          print(t);
   }
}
```

HAPPY CODING!