Assignment # 4

Recursion

Submission Dead Line: 30/10/2022

Implement recursive solutions for problems given below.

1. Reverse Digits:

Returns the positive integer that you get, when you reverse the digits of parameter n. For example, when called with the parameter 12345, this method would return 54321. (Do this with proper integer arithmetic instead of just simply converting to String)

```
int reverseDigits(int n, int & power)
```

2. Even Odd:

It will rearrange an array of int values so that all the even values appear before all the odd values.

```
void evenOdd (int * A, int size);
```

3. Check sum:

It will check if an array A of integers contains an integer A[i] that is the sum of two integers that appear earlier in A, that is, such that A[i] = A [j] + A[k] for j, k < i.

```
bool check_sum (int * A, int size);
```

4. Count Paths:

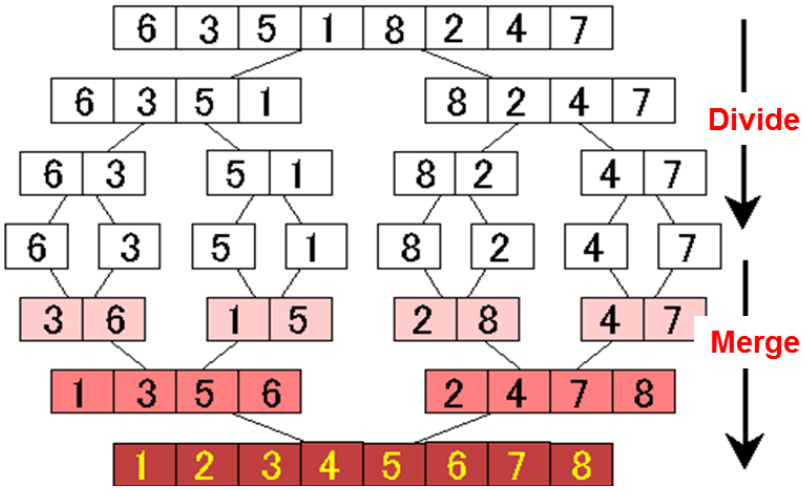
You are standing at the point (n,m) of the grid of positive integers and want to go to origin (0,0) by taking steps either to left or down: that is, from each point (n,m) you are allowed to move either to the point (n-1,m) or the point (n,m-1). Write a method countPaths that counts the number of different paths from the point (n, m) to the origin.

```
int countPaths(int n, int m);
```

5. Merge Sort.

In computer science, merge sort is an O(n log n) comparison-based sorting algorithm.

a. Divide the unsorted list into n sub lists, each containing 1 element (a list of 1 element is considered sorted).
b. Repeatedly merge sub lists to produce new sorted sub lists until there is only 1 sub list remaining. This will be the sorted list.



Input: unsorted integer array, starting and ending indexes

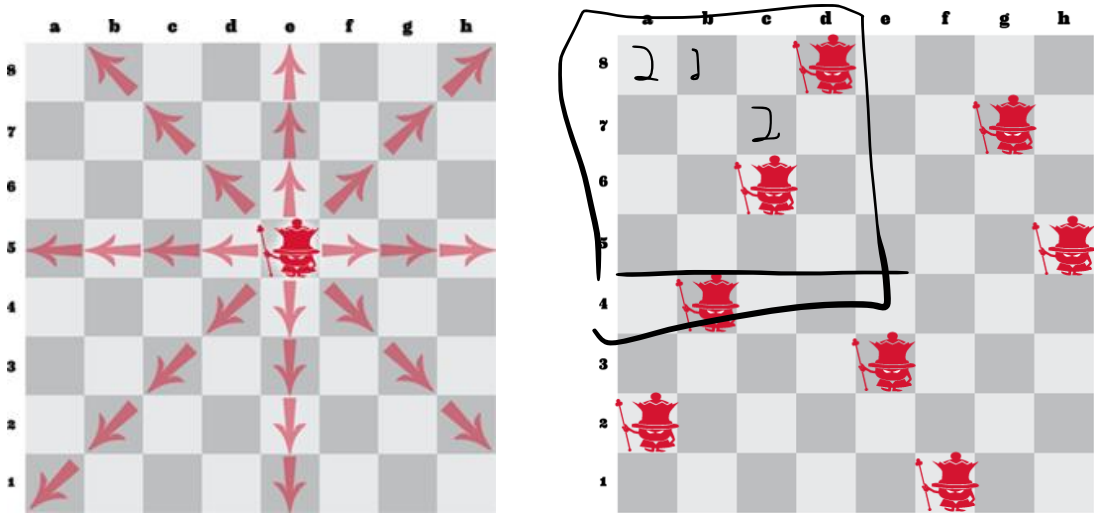
```
void MergeSort(int arr[], int low, int high);
```

6. **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 attack each other. Thus, a solution requires that no two queens share the same row, column, or diagonal.

This function will return an integer array with stored column numbers of queens as solution.

```
int * placeQueens();
```



7. **Find all Subsets of Size k for a given Set**.

The function ComputeSubsets() will take a dynamic array (Set ***S***), size of array and an integer ***k*** (where $1 \geq k \leq$ Set-Size) as parameters and will return a 2-D dynamic array, which contain all subsets of set ***s*** of size ***k***.
For example, let ***S*** = {1, 2, 3, 4, 6} and ***k***=3
then your function must return all subsets of size 3 which will be
{1,2,3}, {1,2,4}, {1,2,6}, {1,3,4}, {1,3,6}, {1,4,6}, {2,3,4}, {2,3,6}, {2,4,6}, {3,4,6}.