DAY 30

1. Merge Sort in Decreasing Order

Description
Write a code that sorts an array in decreasing order using Merge Sort algorithm.

Note that the array in the question which needs to be sorted is named as "randomNumbers". You need to sort the numbers inside the "randomNumbers" array and store the sorted array inside the "sortedNumbers" array.

Input:

1. The first input is 'n', that is, the number of elements in the array.
2. In the next line, n elements of the array space-separated.

Output:
The array sorted in decreasing order.

Sample Test Case-1
Sample Input-1:89 45 76 23 47 1 5 11
Sample Output-1[76,47,45,23,11,9,5,1]
The first input is n (number of elements in the array), here it is 8. The next line in the input represents the elements of the array (space-separated). The output is the same array arranged in decreasing order.

Sample Test Case-2
Sample Input-1:41 4 19 3
Sample Output-1[19,4,3,1]
The first input is n (number of elements in the array), here it is 4. The next line in the input represents the elements of the array (space-separated). The output is the same array arranged in decreasing order.

CODE :

JAVA

```
import java.util.Arrays;
import java.util.Scanner;
public class Source {

    public static int[] sort(int[] randomNumbers) {
        return mergeSort(randomNumbers, 0, randomNumbers.length - 1);
    }

    public static int[] mergeSort(int[] numbers, int first, int last) {
        if (first < last) {
            int mid = (first + last) / 2;
            mergeSort(numbers, first, mid);
            mergeSort(numbers, mid + 1, last);
```

```
            merge(numbers, first, mid, last);
        }

        return numbers;
    }

    private static int[] merge(int[] numbers, int i, int m, int j) {
        int l = i; //inital index of first subarray
        int r = m + 1; // initial index of second subarray
        int k = 0; // initial index of merged array
        int[] t = new int[numbers.length];

        while (l <= m && r <= j) {
            if (numbers[l] >= numbers[r]) {
                t[k] = numbers[l];
                k++;
                l++;
            } else {
                t[k] = numbers[r];
                k++;
                r++;
            }
        }

        // Copy the remaining elements on left half , if there are any
        while (l <= m) {
            t[k] = numbers[l];
            k++;
            l++;
        }

        // Copy the remaining elements on right half , if there are any
        while (r <= j) {
            t[k] = numbers[r];
            k++;
            r++;
        }

        // Copy the remaining elements from array t back the numbers array
        l = i;
        k = 0;
        while (l <= j) {
            numbers[l] = t[k];
            l++;
            k++;
        }

        return numbers;
    }
```

```
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        int size = scanner.nextInt();
        int[] randomNumbers = new int[size];
        for (int i = 0; i < size; i++) {
            randomNumbers[i] = scanner.nextInt();
        }
        int[] sortedNumbers;
        sortedNumbers = sort(randomNumbers);
        System.out.println(Arrays.toString(sortedNumbers));
    }
}
```

2. Quick Sort

Description
Write a program in Java that takes an array of strings as input and returns the sorted array. Assume that the sorting needs to be done based on the size of each string. So, a string with fewer characters should come before another string with more number of characters. Use the quicksort algorithm for the program.

Sample Input - 1:
7
Christene
Tomas
Marline
Marcelluss
Michelle
Quiana
Keny

Sample Output - 1:
Keny
Tomas
Quiana
Marline
Michelle
Christene
Marcelluss
So, the first input is the number of elements in the array. In this case, the number of elements is 7. Next 7 inputs are the elements inside the array. As you can see in the output, the names are sorted as per their length. The name with fewer characters comes before the names with more characters.

Sample Input - 1:
3
Cat
Tree
Bag

Sample Output - 2:
Cat
Bag
Tree

So, the first input is the number of elements in the array. In this case, the number of elements is 3. Next 3 inputs are the elements inside the array. As you can see in the output, the names are sorted as per their length.

Please note that Quick Sort is an **unstable sorting** algorithm. So, the words with same number of characters may come in a different sequence for different people depending upon the factors outside our control.

CODE :

JAVA

```
import java.util.Scanner;
import java.util.Arrays;
 class Source {
     public static String[] quickSort(String array[], int left, int right) {
         if (left < right) {
               int position = partition(array, left, right);
               quickSort(array, left, position - 1);
               quickSort(array, position + 1, right);
         }
      return array;
    }
    public static void print_out(String[] arr){
         int i=0;
         for(i=0;i<arr.length;i++){
               System.out.println(arr[i]);
         }
    }
    public static int partition(String[] arr,int s,int e){
         String pivot_ele=arr[e];
         int i=s;
         for(int j=s;j<e;j++){
               if(arr[j].length()<=pivot_ele.length()){
                     String temp=arr[j];
                     arr[j]=arr[i];
                     arr[i]=temp;
                     i+=1;
               }
         }
         String p_temp=arr[i];
         arr[i]=arr[e];
         arr[e]=p_temp;
         return i;
```

```
    }
    public static void main(String[] args){
        Scanner scanner=new Scanner(System.in);
        int size=scanner.nextInt();
        String[] random_char=new String[size];
        for(int i=0;i<size;i++){
            random_char[i]=scanner.next();
        }
        random_char=quickSort(random_char,0,size-1);
        print_out(random_char);
    }
}
```

3. Sharing Of Chocolate Bar

Description

Rohit and Rahul are very close friends, and as Rahul's birthday is coming up, Rohit decides to share a chocolate bar with him. Rohit has made you the in-charge for the sharing of this chocolate. Each square of the bar has a number written on it, and Rohit has decided that he will share a contiguous piece of chocolate, in which the sum of each number on the squares is equal to the month in which Rahul was born, and the numbers written on these squares add up to Rahul's birth date. You need to find the number of ways the chocolate bar can be divided.

Input: The input should be in the following format:
The first line should be an integer 'n' representing the size of the chocolate bar.
The second line should be 'n' elements of array space separated.
Each element of the array indicates the number on the square of the chocolate bar.
The third line should be the value of d indicating the date of Rahul's birthday.
The fourth line should be the value of m indicating the month of Rahul's birthday.

Output: The output should be in the following format:
Print the number of ways the chocolate bar can be divided.

Sample test case:
Input:
61 3 5 2 4 514 4
Output:
1

CODE :

JAVA

```
import java.util.*;

public class Source {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
```

```
        int n = s.nextInt();
        int[] arr = new int[n];
        for(int i=0; i < n; i++){
            arr[i] = s.nextInt();
        }
        int d = s.nextInt();
        int m = s.nextInt();
        numberOfSubArrays(arr, n, d, m);
    }
    // Method to find number of subarrays of size m whose sum of elements is d
    static void numberOfSubArrays(int[] arr,int n, int d, int m){
        int c=0;
        int i;
        int birth_sums=0;
        if(n<m){
            System.out.println(0);
            return ;
        }
        for(i=0;i<m;i++){
            birth_sums+=arr[i];
        }
        if(birth_sums==d){
            c+=1;
        }
        i=0;
        for(int j=m;j<n;j++){
            birth_sums=birth_sums-arr[i]+arr[j];
            if(birth_sums==d){
                c+=1;
            }
            i+=1;
        }
        System.out.println(c);
    }
}
```