CDAC-Mumbai

# Subject: Coding Challenge

**Date:18-09-2024 Total Marks:5**

# Time duration: 1:30hrs. Batch: Aug-24

**----------------------------------------------------------------------------------------------------------------**

1. Write a Java program to prove that Euclid’s algorithm computes the greatest common divisor of two positive given integers. **[1 Marks]**

*"The Euclidean algorithm is based on the principle that the greatest common divisor of two numbers does not change if the larger number is replaced by its difference with the smaller number. For example, 21 is the GCD of 252 and 105 (as 252 = 21 × 12 and 105 = 21 × 5), and the same number 21 is also the GCD of 105 and 252 − 105 = 147. Since this replacement reduces the larger of the two numbers, repeating this process gives successively smaller pairs of numbers until the two numbers become equal. When that occurs, they are the GCD of the original two numbers. By reversing the steps, the GCD can be expressed as a sum of the two original numbers each multiplied by a positive or negative integer, e.g., 21 = 5 × 105 + (−2) × 252. The fact that the GCD can always be expressed in this way is known as Bézout's identity."*

import java.util.Scanner;

public class EuclideanGCD {

public static int gcd(int a, int b) {

// Base case: if b becomes 0, return a as GCD

if (b == 0) {

return a;

}

// Recursive case: GCD of (b, a % b)

return gcd(b, a % b);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the first number:");

int num1 = sc.nextInt();

System.out.println("Enter the second number:");

int num2 = sc.nextInt();

int result = gcd(num1, num2);

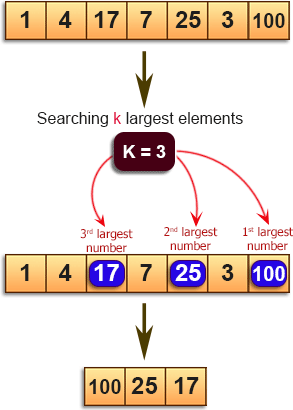
System.out.println("The GCD of " + num1 + " and " + num2 + " is: " + result);

}

}

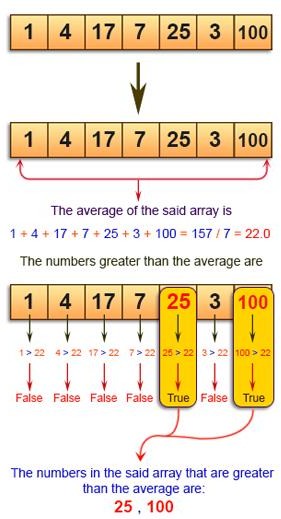
`**2.** Write a Java program to find the k largest elements in a given array. Elements in the array can be in any order (Use scanner class to input and Array, below picture is example do not make static array as repsrented in example). **[1 Mark]**

# Pictorial Presentation:



1. Write a Java program to find the numbers greater than the average of the numbers of a given array (Use scanner class to input and Array, below picture is example do not make static array as repsrented in example). **[1 mark]**

# Pictorial Presentation:



import java.util.Scanner;

public class NumberGreater{

    public static void main(String[] args){

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter number of elements");

        int n=sc.nextInt();

        int[] numbers=new int[n];

        System.out.println("Enter the elements of an Array: ");

        for(int i=0;i<n;i++){

            numbers[i]=sc.nextInt();

        }

        double sum=0;

        for (int number:numbers){

            sum+=number;

        }

        double average=sum/n;

        System.out.println("The average is :"+average);

        System.out.println("Number greater than average:");

        for(int number:numbers){

            if(number>average){

                System.out.println(number);

            }

            sc.close();

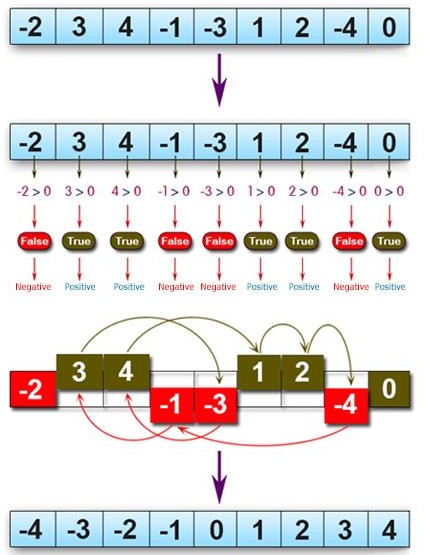
        }

        }

    }

Q4. Write a Java program to move every positive number to the right and every negative number to the left of a given array of integers (Use scanner class to input and Array, below picture is example do not make static array as repsrented in example). **[1 Mark]**

# Pictorial Presentation:



import java.util.Scanner;

public class MoveNegativesLeftPositivesRight {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the number of elements in the array: ");

        int n = scanner.nextInt();

    int[] arr = new int[n];

        System.out.println("Enter the elements of the array (both positive and negative):");

        for (int i = 0; i < n; i++) {

            arr[i] = scanner.nextInt();

        }

          rearrangeArray(arr);

     System.out.println("Rearranged array:");

           for (int num : arr) {

       System.out.print(num + " ");

        }

    }

    public static void rearrangeArray(int[] arr) {

          int left = 0;

        int right = arr.length - 1;

        while(left <= right) {

        if (arr[left] < 0) {

                  left++;

            }

              else if (arr[right] >= 0) {

                right--;

            }

             else {

                 int temp = arr[left];

                arr[left] = arr[right];

                 arr[right] = temp;

                left++;

                right--;

            }

        }

    }

}

5. Write a Java program to find the median of the number inside the window (size k) at each moving in a given array of integers with duplicate numbers. Move the window from the start of the array. **[1 Mark]**

Example:

{|1, 2, 3|, 4, 5, 6, 7, 8, 8} -> Return median 2

{1, |2, 3, 4|, 5, 6, 7, 8, 8} -> Return median 3

{1, 2, |3, 4, 5|, 6, 7, 8, 8} -> Return median 4

{1, 2, 3, |4, 5, 6|, 7, 8, 8} -> Return median 5

{1, 2, 3, 4, |5, 6, 7|, 8, 8} -> Return median 6

{1, 2, 3, 4, 5, |6, 7, 8|, 8} -> Return median 7

{1, 2, 3, 4, 5, 6, |7, 8, 8|} -> Return median 8

Result array {2, 3, 4, 5, 6, 7, 8}

Note: Above representation is just example, use scanner class for input the array and for getting size of window to slide.

import java.util.Arrays;

import java.util.Scanner;

public class SlidingWindowMedian {

public static double findMedian(int[] window) {

Arrays.sort(window); // Sort the window

int n = window.length;

if (n % 2 == 1) { // Odd number of elements, median is the middle one

return window[n / 2];

} else { // Even number of elements, median is the average of the two middle ones

return (window[(n / 2) - 1] + window[n / 2]) / 2.0;

}

}

public static void findSlidingWindowMedian(int[] arr, int k) {

for (int i = 0; i <= arr.length - k; i++) {

int[] window = Arrays.copyOfRange(arr, i, i + k);

System.out.println("Window: " + Arrays.toString(window) + " -> Median: " + findMedian(window));

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the array:");

int n = sc.nextInt();

int[] arr = new int[n];

System.out.println("Enter the elements of the array:");

for (int i = 0; i < n; i++) {

arr[i] = sc.nextInt();

}

System.out.println("Enter the size of the window (k):");

int k = sc.nextInt();

findSlidingWindowMedian(arr, k);

}

}