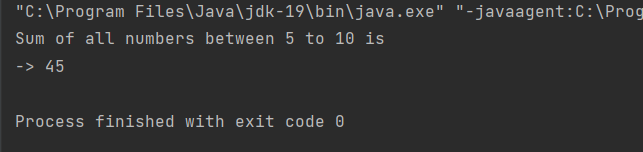
22K-5159 **Amna Mansoor** BSE-3B LAB-5

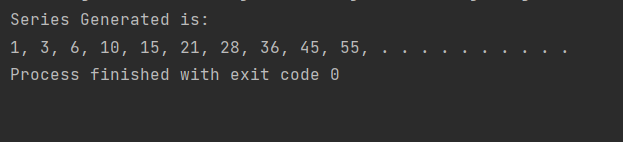
***Task 1:***

public class Task1 {  
  
 //Task#1 Use recursion to add all of the numbers between 5 to 10.  
  
 public static void main(String[] args) {  
 int a=5;  
 int b=10;  
 int sum=*Add*(a,b);  
 System.*out*.println("Sum of all numbers between 5 to 10 is\n-> "+sum);  
  
 }  
 static int Add(int a, int b){  
 if (a==b){  
 return a;  
 }else return a+*Add*(a+1, b);  
 }  
}



***Task 2-a:***

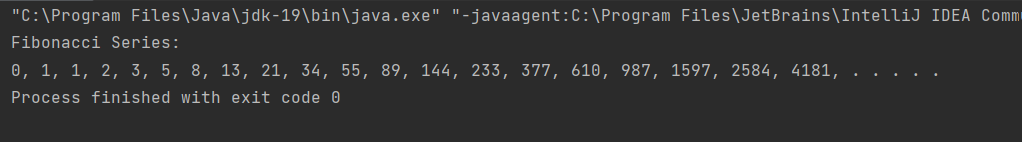
public class Task2\_a {  
  
 //Task-2 (a) Generate the following sequence with recursive approach  
 //1 , 3 , 6 , 10 , 15 , 21 , 28 . . . .  
  
 public static void main(String[] args) {  
 int n = 10;  
 System.*out*.println("Series Generated is:");  
 *Sequence*(1, 2, n);  
  
 }  
  
 public static void Sequence(int current, int increment, int n) {  
 if (n <= 0) {  
 return;  
 }  
 System.*out*.print(current + ", ");  
 *Sequence*(current + increment, increment + 1, n - 1);  
 System.*out*.print(". ");  
 }  
}



***Task 2-b:***

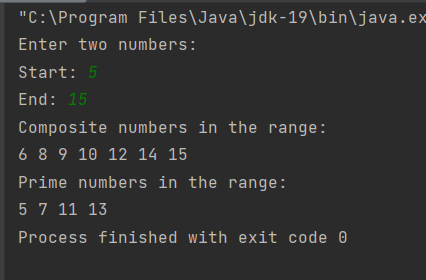
public class Task2\_b {  
  
 //Task-2 b. Generate the following sequence with recursive approach  
 //0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 , 89 , 144 . . .  
  
 public static void main(String[] args) {  
 int n=20;  
 System.*out*.println("Fibonacci Series:");  
 for (int i=0;i<n;i++){  
 System.*out*.print(*Fibonacci*(i)+", ");  
 }  
 System.*out*.print(". . . . . ");  
 }  
 public static int Fibonacci(int n){  
 if (n<=1){  
 return n;  
 }  
 return *Fibonacci*(n-1)+*Fibonacci*(n-2);  
  
  
 }

}



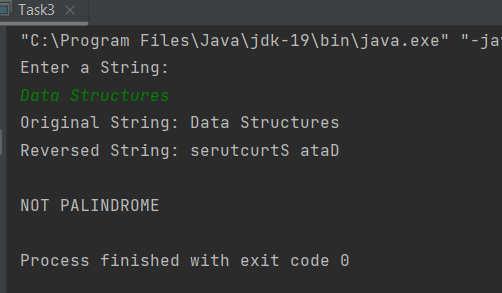
***Task#2:***

import java.util.Scanner;  
  
public class Task2 {  
  
 //Task#2 Write a recursive function that takes to integer as a parameters and finds out all composite  
 //and prime number in given range.  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter two numbers:");  
 System.*out*.print("Start: ");  
 int start = sc.nextInt();  
 System.*out*.print("End: ");  
 int end = sc.nextInt();  
  
 System.*out*.println("Composite numbers in the range:");  
 *Composite*(start, end, start);  
  
 System.*out*.println("\nPrime numbers in the range:");  
 *Prime*(start, end, start);  
 }  
  
 public static boolean isPrime(int n,int divisor) {  
 if(n<=2) {  
 return (n==2);  
 }  
 if(n % divisor==0) {  
 return false;  
 }  
 if (divisor\*divisor>n) {  
 return true;  
 }  
 return *isPrime*(n,divisor+1);  
 }  
  
 public static void Composite(int start, int end, int current) {  
 if (current <= end) {  
 if (!*isPrime*(current, 2)) {  
 System.*out*.print(current + " ");  
 }  
 *Composite*(start, end, current + 1);  
 }  
 }  
  
 public static void Prime(int start, int end, int current) {  
 if (current <= end) {  
 if (*isPrime*(current, 2)) {  
 System.*out*.print(current + " ");  
 }  
 *Prime*(start, end, current + 1);  
 }  
 }  
}



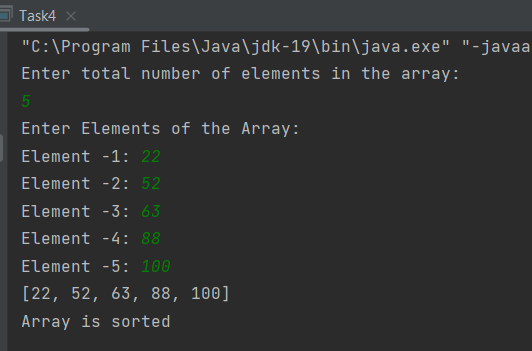
***Task 3:***

import java.util.Scanner;  
  
public class Task3 {  
  
 //Write down the program that reverse the string if reverses string and original string is same  
 //then it prints the “PALINDROM” otherwise “Not PALINDROM”.  
  
 public static void main(String[] args) {  
 System.*out*.println("Enter a String: ");  
 Scanner sc=new Scanner(System.*in*);  
 String str=sc.nextLine();  
 sc.close();  
  
 String reversed=*reverseString*(str);  
 System.*out*.println("Original String: "+str);  
 System.*out*.println("Reversed String: "+reversed);  
  
 if (str.equals(reversed)){  
 System.*out*.println("\nPALINDROME");  
 }else{  
 System.*out*.println("\nNOT PALINDROME");  
 }  
 }  
  
 public static String reverseString(String str) {  
 if(str==null || str.isEmpty()){  
 return str;  
 }  
 else {  
 return *reverseString*(str.substring(1) )+ str.charAt(0);  
 }  
 }  
}



***Task 4:***

import java.util.Arrays;  
import java.util.Scanner;  
  
public class Task4 {  
  
 //Task#4: Write a recursive method in java that check that given array is sorted or not.  
  
 public static void main(String[] args) {  
 System.*out*.println("Enter total number of elements in the array: ");  
 Scanner sc=new Scanner(System.*in*);  
 int n=sc.nextInt();  
 int[] array=new int[n];  
  
 System.*out*.println("Enter Elements of the Array: ");  
 for (int i=0;i<n;i++){  
 System.*out*.print("Element -"+(i+1)+": ");  
 array[i]=sc.nextInt();  
 }  
 System.*out*.println(Arrays.*toString*(array));  
 if(*isSorted*(array,n)){  
 System.*out*.println("Array is sorted");  
 }else{  
 System.*out*.println("Array is not sorted");  
 }  
 }  
 public static boolean isSorted(int[] array,int n) {  
 if(n<=1){  
 return true;  
 }  
 if(array[n-1]<array[n-2]){  
 return false;  
 }  
 return *isSorted*(array,(n-1));  
 }  
}



***Task 5:***

import java.util.ArrayList;  
import java.util.List;  
  
public class Task5 {  
  
 // Task#5: Write a program that finds the Subset of targeted sum.  
  
 public static void main(String[] args) {  
 int[] array = {1, 2, 3, 4};  
  
 System.*out*.println("TARGET = 7");  
 System.*out*.println("Subsets derived from Targeted sum: ");  
 boolean foundSubsets1 = *Subsets*(array, 7, 0, new ArrayList<>());  
 if (!foundSubsets1) {  
 System.*out*.println("No subsets");  
 }  
  
 System.*out*.println("\nTARGET = 11");  
 System.*out*.println("Subsets derived from Targeted sum: ");  
 boolean foundSubsets2 = *Subsets*(array, 11, 0, new ArrayList<>());  
 if (!foundSubsets2) {  
 System.*out*.println("No subsets");  
 }  
 }  
  
 public static boolean Subsets(int[] array, int target, int pos, List<Integer> ListSubset) {  
 if (target == 0) {  
  
 System.*out*.println(ListSubset);  
 return true;  
 }  
 if (pos == array.length) {  
 return false;  
 }  
 ListSubset.add(array[pos]);  
 boolean foundSubsets = *Subsets*(array, target - array[pos], pos + 1, ListSubset);  
 ListSubset.remove(ListSubset.size() - 1);  
 foundSubsets = foundSubsets || *Subsets*(array, target, pos + 1, ListSubset);  
 return foundSubsets;  
 }  
}

