**WEEK 7**

1# Write a Java function to implement binary search.

**CODE:**

package week\_7;

import java.util.Scanner;

public class BinarySearch {

public static int BinarySearch(int[] arr,int start,int end,int key){

if (start>end)

return -1;

else{

int middle = (start+end)/2;

if (arr[middle]==key)

return middle;

if (arr[middle]>key)

return BinarySearch(arr,start,middle-1,key);

return BinarySearch(arr,middle+1,end,key);

}

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

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

int n=sc.nextInt();

int[] arr=new int[n];

System.out.println("Enter elements (must be sorted): ");

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

arr[i]=sc.nextInt();

}

System.out.println("Enter key: ");

int key=sc.nextInt();

int result=BinarySearch(arr,0,n-1,key);

if (result==-1){

System.out.println(key+" is not in array");

}

else{

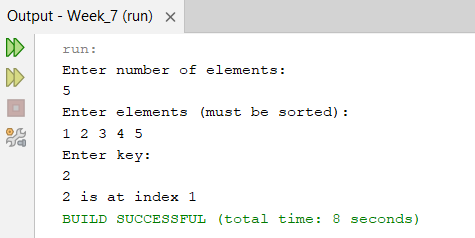
System.out.println(key + " is at index " + result);

}

}

}

**OUTPUT:**

****

2# Write a Java function to arrange the elements of an array in ascending order (Sorting).

**CODE:**

package week\_7;

import java.util.Scanner;

public class Sorting {

public static void BubbleSort(int[] arr){

int len=arr.length;

for (int i=0;i<len-1;i++){

for(int j=0;j<len-1-i;j++){

if (arr[j]>arr[j+1]){

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

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

int n=sc.nextInt();

int[] arr=new int[n];

System.out.println("Enter elements: ");

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

arr[i]=sc.nextInt();

}

BubbleSort(arr);

System.out.println("Sorted array is: ");

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

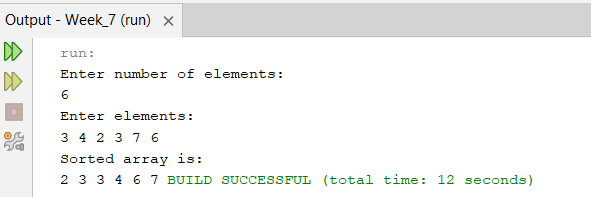
System.out.print(arr[i] + " ");

}

}

}

**OUTPUT:**



3# Write a program to reverse a given string.

**CODE:**

package week\_7;

import java.util.Scanner;

public class ReverseString {

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

System.out.println("Enter string: ");

String s=sc.nextLine();

String rev="";

int len=s.length();

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

char c=s.charAt(i);

rev=c+rev;

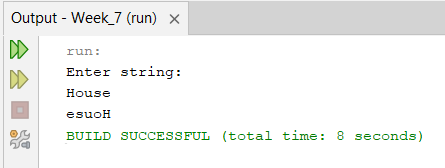
}

System.out.println(rev);

}

}

**OUTPUT:**



4# Write a program to check whether a given string is palindrome or not.

**CODE:**

package week\_7;

import java.util.Scanner;

public class PalindromeString {

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

System.out.println("Enter string: ");

String s=sc.nextLine();

String rev="";

int len=s.length();

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

char c=s.charAt(i);

rev=c+rev;

}

if (s.equals(rev)){

System.out.println("Plaindrome");

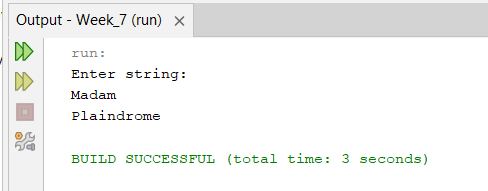
} else{

System.out.println("Not a Plaindrome");

}

} }

**OUTPUT:**



5# Write a program to implement factorial of a number through recursion.

**CODE:**

package week\_7;

import java.util.Scanner;

public class Factorial {

public static int fact(int n){

if (n==1)

return 1;

return n\*fact(n-1);

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

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

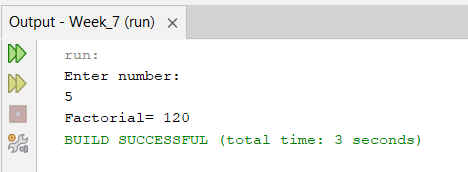
int n=sc.nextInt();

System.out.println("Factorial= " + fact(n));

}

}

**OUTPUT:**

****

6# Write a program to implement Fibonacci series of a number with and without recursion.

**CODE:**

package week\_7;

import java.util.Scanner;

public class Fibonacci {

public static int fibo(int n){

if (n<=1)

return n;

return fibo(n-1)+fibo(n-2);

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

System.out.println("Enter number of terms: ");

int n=sc.nextInt();

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

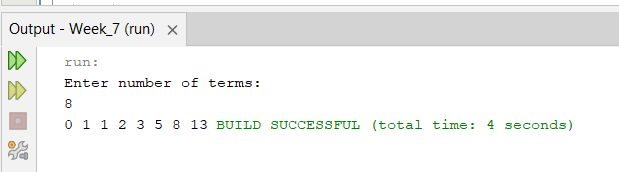
System.out.print(fibo(i) + " ");

}

}

}

**OUTPUT:**

****

**Optional:**

7# Write a Java function to find the greatest common divisor (GCD) of two numbers with and without using recursion.

**CODE:**

package week\_7\_optional;

import java.util.Scanner;

public class GCD {

public static int gcdIterative(int x, int y){

int r=x%y;

while(r>0){

x=y;

y=r;

r=x%y;

}

return y;

}

public static int gcdRecursive(int x,int y){

if (y==0)

return x;

return gcdRecursive(y,x%y);

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

System.out.println("Enter two numbers: ");

int x=sc.nextInt();

int y=sc.nextInt();

System.out.println("GCD By iterative method: ");

System.out.println(gcdIterative(x,y));

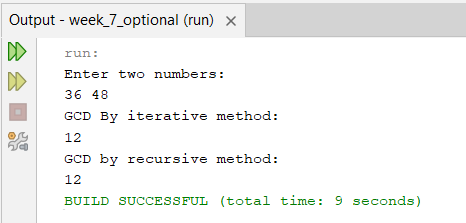
System.out.println("GCD by recursive method: ");

System.out.println(gcdRecursive(x,y));

}

}

**OUTPUT:**

****

8# Write a program to check whether two strings are anagrams of each other (“listen” and “silent” are anagrams).

**CODE:**

package week\_7\_optional;

import java.util.Scanner;

public class Anagrams {

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

System.out.println("Enter first string: ");

String s1=sc.nextLine().toLowerCase();

System.out.println("Enter Second string: ");

String s2=sc.nextLine().toLowerCase();

int[] count=new int[256];

s1 = s1.replaceAll("\\s", "");

s2 = s2.replaceAll("\\s", "");

if(s1.length()!=s2.length()){

System.out.println("Not anagrams");

return;

}

for(int i=0;i<s1.length();i++){

count[s1.charAt(i)]++;

}

for(int i=0;i<s2.length();i++){

count[s2.charAt(i)]--;

}

for(int i:count){

if (i!=0){

System.out.println("Not anagrams");

return;

}

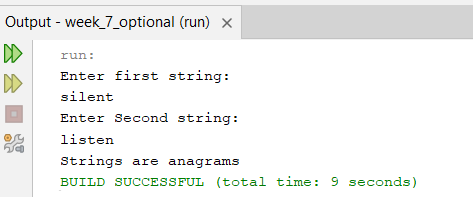
}

System.out.println("Strings are anagrams");

}

}

**OUTPUT:**

****

9# Implement quick sort using recursion.

**CODE:**

package week\_7\_optional;

import java.util.Scanner;

public class quickSort {

public static void quicksort(int[] arr,int low,int high){

if (low<high){

int j= partition(arr,low,high);

quicksort(arr,low,j-1);

quicksort(arr,j+1,high);

}

}

public static int partition(int[] arr,int low,int high){

int pivot=arr[low];

int i=low+1;

int j=high;

while(i<j){

while(i <= j && arr[i]<=pivot)i++;

while(i <= j && arr[j]>=pivot)j--;

if(i>=j)break;

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

arr[low]=arr[j];

arr[j]=pivot;

return j;

}

public static void main(String[] args) {

Scanner sc= new Scanner (System.in);

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

int n=sc.nextInt();

int[] arr=new int[n];

System.out.println("Enter elements: ");

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

arr[i]=sc.nextInt();

}

quicksort(arr,0,n-1);

System.out.println("Sorted array is: ");

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

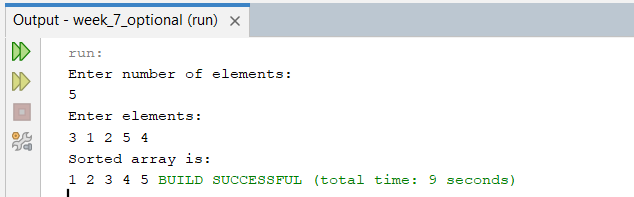
System.out.print(arr[i] + " ");

}

}

}

**OUTPUT:**

****