Name : D.Dharma Teja

Roll no: 22011F0027(C.S.E.)

**JAva Programming**

Assignment

1. **The Fibonacci sequence is defined by the following rule: The fist two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.**

**Code:**

import java.util.Scanner;

public class Fibonacci {

    static void fibonacci(int n){

        int a=0,b=1,c=0;

        for (int i=0;i<n;i++){

            a=b;

            b=c;

            c=a+b;

        }

        System.out.println(c);

    }

    static int recFibonaci(int n){

        if(n==1||n==2){

            return 1;

        }

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

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("enter the n value");

        int n = sc.nextInt();

        System.out.print( "term no. " + n +" WITHOUT using Recursion :");

        fibonacci(n);

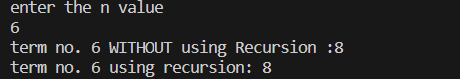
        System.out.println("term no. "+ n+ " using recursion: "+recFibonaci(n));

        sc.close();

    }

}

**Output:**

****

1. **Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.**

**Code:**

import java.util.\*;

public class prime {

    public static void printPrime(int n){

        for(int i=1;i<=n;i++){

            int count=0;

            for(int j=1;j<=i;j++){

                if(i%j==0){

                    count++;

                }

            }

            if(count==2){

               System.out.println(i);

            }

        }

    }

    public static void main(String[] args) {

        Scanner input=new Scanner(System.in);

        System.out.println("enter the n value");

        int n =input.nextInt();

        input.close();

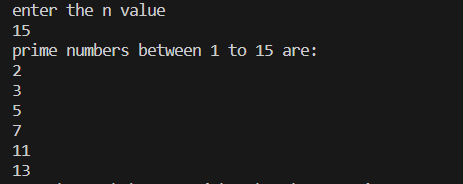
        System.out.println("prime numbers between 1 to "+n+" are:");

        printPrime(n);

    }

}

**Output:**



3**. Implement the complex number ADT in Java using a class. The complex ADT is used to represent complex numbers of the form c = a + ib, where a and b are real numbers. The operations supported by this ADT are:**

**a)Reading a complex number. b)Writing a complex number. c)Addition of two complex numbers.d)Multiplication of two complex numbers.**

**Code:**

import java.util.\*;

public class Complex {

    int real,img;

    void readComplex() {

        Scanner input =new Scanner(System.in);

        System.out.println("enter the real part ");

        this.real=input.nextInt();

        System.out.println("enter the imaginary part ");

        this.img=input.nextInt();

    }

    void writeComplex(){

        if(this.img>0)

            System.out.println(this.real+"+"+this.img+"i");

        else

            System.out.println(this.real+" "+this.img+"i");

    }

    void addComplex(Complex c){

        Complex result=new Complex();

        result.real=this.real+c.real;

        result.img=this.img+c.img;

        System.out.println("after addition:");

        result.writeComplex();

    }

    void multiplyComplex(Complex c){

        Complex result =new Complex();

        result.real=this.real\*c.real-(this.img\*c.img);

        result.img=this.real\*c.img+this.img+c.real;

        System.out.println("after multiplication :");

        result.writeComplex();

    }

    public static void main(String[] args) {

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

        Complex c1 = new Complex();

        c1.readComplex();

        System.out.println("second number");

        Complex c2=new Complex();

        c2.readComplex();

        Scanner Sc = new Scanner(System.in);

        int choice;

        do {

            System.out.println("enter\n1.for Addition\n2.for multiplication\n3. To exit");

            choice=Sc.nextInt();

            switch(choice){

                case 1:c1.addComplex(c2);

                        break;

                case 2:c1.multiplyComplex(c2);

                        break;

                default:

                        Sc.close();

                        System.exit(0);

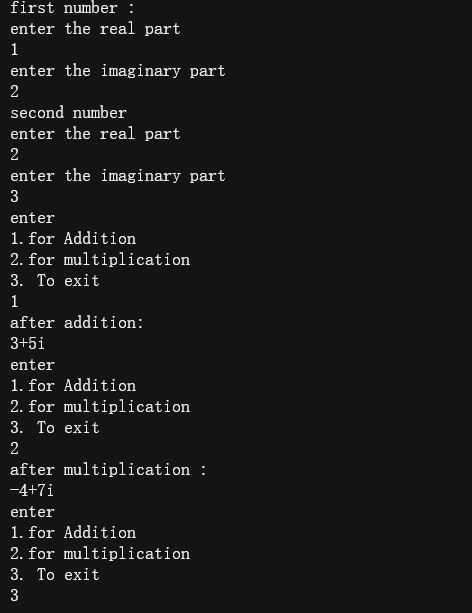
            }

        }while(true);

    }

}

**Output:**

**4. Write a Java program to illustrate method overloading.(Ex. Write functions for finding the maximum of two and three numbers**.

**Code:**

import java.util.\*;

public class MethodOverloading {

    public static void main(String[] args) {

        Scanner sc =new Scanner(System.in);

        System.out.println("enter the values ");

        int a=sc.nextInt(),b=sc.nextInt(),c=sc.nextInt();

       System.out.println(max.maximum(a, b)+"is maximum of "+a+" "+b);

       System.out.println(max.maximum(a, c)+"is maximum of "+a+" "+c);

       System.out.println(max.maximum(b, c)+"is maximum of "+b+" "+c);

       System.out.println(max.maximum(a, b, c)+"is maximum among all the 3 numbers");

       sc.close();

    }

}

class max{

    static int maximum(int a, int b){

      int max= a>b?a:b;

      return max;

    }

    static int maximum(int a,int b, int c){

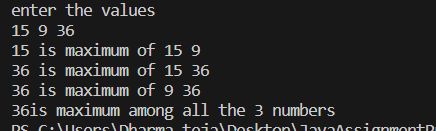
        int max=a>b&&a>c?a:b>c&&b>a?b:c;

        return max;

    }

}

**Output:**

****

**5. Write a Java program to find both the largest and smallest number in a list of numbers.**

**Code:**

import java.util.\*;

public class MInMax {

    static void minMax(int arr[]){

        int min=arr[0],max=arr[0];

        for(int i=1;i<arr.length;i++){

            if(arr[i]<min){

                min=arr[i];

            }

            if(arr[i]>max){

                max=arr[i];

            }

        }

        System.out.println("minimum element in the given array is \'"+min+"\'' maximum element is \'"+max+"\'");

    }

    public static void main(String[] args) {

        Scanner input =new Scanner(System.in);

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

        int size=input.nextInt();

        int[] a=new int[size];

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

        for(int i=0;i<size;i++){

            a[i]=input.nextInt();

        }

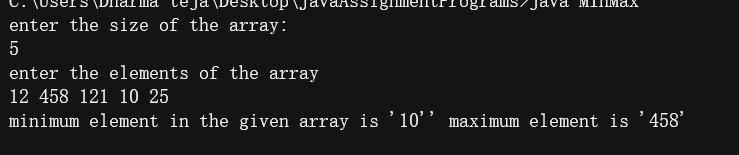
        minMax(a);

        input.close();

    }

}

**Output:**



**6.** **Write a Java program that uses both recursive and non recursive linear search functions to search for a character key in a list of characters.**

**Code:**

import java.util.Scanner;

public class LinearSearch{

    static void recLinearSearch(char []ar,char key,int index){

        if(index>=ar.length){

            System.out.println("element not found ");

            return;

        }

        if (ar[index]==key){

            System.out.println("found at: "+index);

            return;

        }

        recLinearSearch(ar, key, index+1);

    }

    static void linearSearch(char []a ,char key){

        int i=0;

        boolean found=false;

        while(!found&&i<a.length){

            if(key ==a[i]){

                found =true;

                System.out.println("element found at index: "+i );

            }

            i++;

        }

        if(found==false){

            System.out.println("element not found");

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int size = sc.nextInt();

        char [] arr=new char[size];

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

        for(int i=0;i<size;i++){

            arr[i]=sc.next().charAt(0);

        }

        System.out.println("enter the key element to be searched:");

        char key = sc.next().charAt(0);

        // recLinearSearch(arr, key, 0);

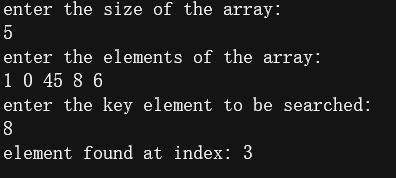
        linearSearch(arr, key);

        sc.close();

    }

}

**Output:**



**7. Write a Java program to implement the matrix ADT using a class. The operations supported by this ADT are:**

**a)Reading a matrix. b)Addition of two matrices. c)Printing a matrix.**

**d)Multiplication of two matrices**.

**Code:**

import java.util.\*;

public class Matrix {

    //member variables

    int row,column;

    int[][] matrix;

    //constructors

    Matrix(){

        row=0;column=0;

        matrix = new int[row][column];

    }

    Matrix(int a,int b){

        row=a;column=b;

        matrix=new int[row][column];

    }

    //member functions

    void readMatrix(){

        Scanner sc = new Scanner(System.in);

        System.out.println("enter the elements of the matrix:");

        for(int i=0;i<this.row;i++){

            for(int j=0;j<this.column;j++){

                this.matrix[i][j]=sc.nextInt();

            }

        }

    }

    void printMatrix(){

        for(int i=0;i<this.row;i++){

            for(int j=0;j<this.column;j++){

                System.out.print(this.matrix[i][j]+" ");

            }

            System.out.println();

        }

    }

    Matrix addMatrix(Matrix a){

        Matrix result=null;

        if(this.row==a.row&&this.column==a.column){

            result= new Matrix(this.row,this.column);

            for(int i=0;i<this.row;i++){

                for(int j=0;j<this.column;j++){

                    result.matrix[i][j]=this.matrix[i][j]+a.matrix[i][j];

                }

            }

        }else{

            System.out.println("addition is not possible");

        }

        return result;

    }

    Matrix multiplyMatrix(Matrix a) {

        Matrix result = null;

        if(this.column==a.row){

            result = new Matrix(this.row, a.column);

            for(int i=0;i<this.row;i++){

                for(int j=0;j<a.column;j++){

                    for(int k=0;k<a.row;k++){

                        result.matrix[i][j]+=this.matrix[i][k]+a.matrix[k][j];

                    }

                }

            }

        }else{

            System.out.println("matrix multiplication is NOT possible!!");

        }

        return result;

    }

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.println("enter the order of matrix 1");

        int r1=sc.nextInt(),c1=sc.nextInt();

        Matrix m1=new Matrix(r1,c1);

        System.out.println("enter the order of matrix 2");

        int r2=sc.nextInt(),c2=sc.nextInt();

        Matrix m2= new Matrix(r2,c2);

        System.out.println("for matrix 1");

        m1.readMatrix();

        System.out.println("for matrix 2");

        m2.readMatrix();

        do{

            System.out.println("1.Add \n2. multiply \nenter your choice\n Hit any other key to exit");

            int choice=sc.nextInt();

            switch(choice){

                case 1:

                        System.out.println("after adding ");

                        m1.addMatrix(m2).printMatrix();

                        break;

                case 2:

                        System.out.println("after multiplication");

                        m1.multiplyMatrix(m2).printMatrix();

                        break;

                default:

                        sc.close();

                        System.exit(0);

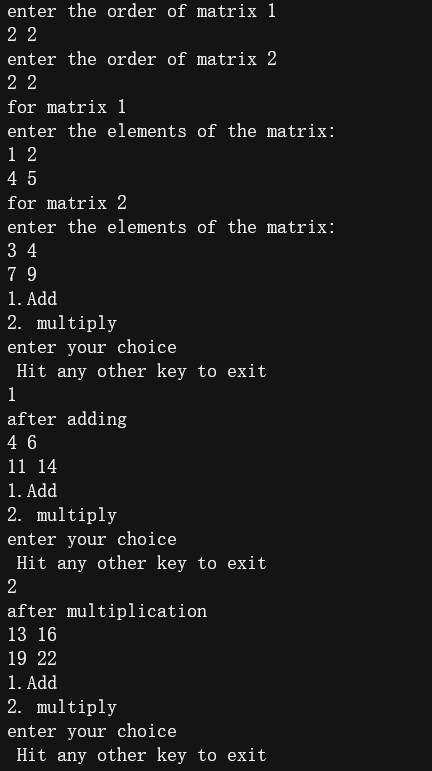
            }

        }while(true);

    }

}

**Output:**

****

8**. Write a Java program that uses functions to perform the following:**

**i)sorts a list of integers in ascending order using bubble sort.**

**ii)then searches for a key value(integer) non recursively in the above sorted list using binary search.**

**Code:**

import java.util.Scanner;

public class SortAndSearch {

public static void bSort(int []ar){

for(int i=0;i<ar.length-1;i++){

for(int j=0;j<ar.length-i-1;j++){

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

int temp = ar[j];

ar[j]=ar[j+1];

ar[j+1]=temp;

}

}

}

}

static void printArray(int[] arr){

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

System.out.println(arr[i]);

}

}

static void bSearch(int []a,int key){

int high=a.length-1,low=0;

boolean found=false;

while(low<=high&&found!=true){

int mid=(high+low)/2;

if(key==a[mid]){

System.out.println("found at index:"+mid);

found = true;

break;

}

if(key<a[mid]){

high=mid-1;

}else if(key>mid){

low=mid+1;

}

}

if(!found){

System.out.println("element not found");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int size = sc.nextInt();

int []arr=new int[size];

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

for(int i=0;i<size;i++)

arr[i]=sc.nextInt();

System.out.println("before sorting:");

printArray(arr);

bSort(arr);

System.out.println("after soring :");

printArray(arr);

System.out.println("enter the key element to be searched:");

int key=sc.nextInt();

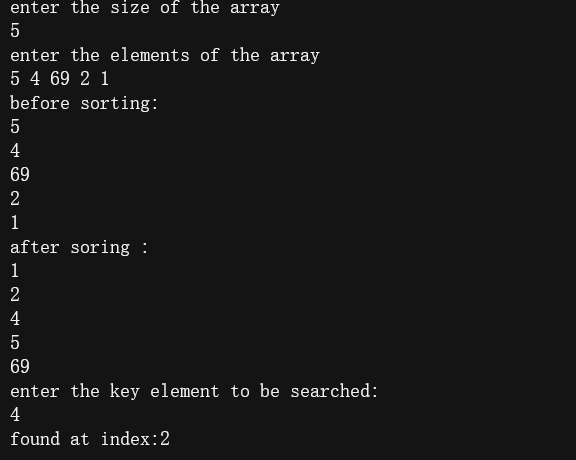
bSearch(arr, key);

sc.close();

}

}

**Output:**

****

**9. Write a Java program that uses functions to perform the following:**

**i)sorts a list of integers in ascending order using insertion sort.**

**ii)then searches for a key value(integer) recursively in the above sorted list**

**using binary search.**

**Code:**

import java.util.Scanner;

public class InsSortandBinSearch {

static void insSort(int [] arr){

for(int i=1;i<=arr.length-1;i++){

int temp = arr[i];

int j=i-1;

while(j>=0&&temp<arr[j]){

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

j--;

}

arr[j+1]=temp;

}

}

static void recBinSearch(int[] arr,int key,int min,int max){

int mid= (min+max)/2;

if(key==arr[mid]){

System.out.println("found at index:"+mid);

return;

}

if(min>max){

System.out.println("element not found!!");

return;

}

if(key<arr[mid]){

recBinSearch(arr, key, 0, mid-1);

}else if(key>mid){

recBinSearch(arr, key, mid+1, max);

}

}

static void printArray(int[] arr){

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

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

}

System.out.println();

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int size = sc.nextInt();

int[] ar= new int[size];

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

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

ar[i]=sc.nextInt();

}

System.out.println("before sort");

printArray(ar);

insSort(ar);

System.out.println();

System.out.println("after sort:");

printArray(ar);

System.out.println("enter the key element to be searched:");

int key =sc.nextInt();

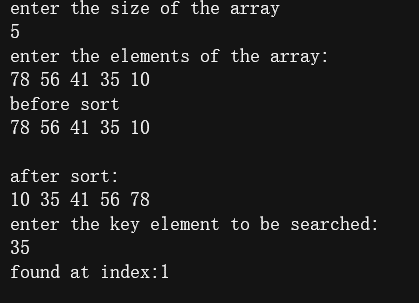
recBinSearch(ar, key, 0, ar.length-1);

sc.close();

}

}

**Output:**



**10. Write a Java program that uses functions to perform the following:**

**i)sorts a list of integers in ascending order using selection sort.**

**ii)then searches for a key value(integer) non recursively in the above sorted list using binary search.** **(SelectionSort.java)**

**Code:**

import java.util.Scanner;

public class SelectionSort {

static void selectionSort(int[] arr){

for(int i=0;i<arr.length-1;i++){

int min=i;

for(int j=i+1;j<arr.length;j++){

if(arr[j]<arr[min]){

min=j;

}

int temp= arr[min];

arr[min]=arr[i];

arr[i]=temp;

}

}

}

static void printArray(int[] arr){

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

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

}

System.out.println();

}

static void BinSearch(int a[],int key){

int high=a.length-1,low=0;

boolean found=false;

while(low<=high&&found!=true){

int mid=(high+low)/2;

if(key==a[mid]){

System.out.println("found at index:"+mid);

found = true;

break;

}

if(key<a[mid]){

high=mid-1;

}else if(key>mid){

low=mid+1;

}

}

if(!found){

System.out.println("element not found");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int size = sc.nextInt();

int []arr=new int[size];

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

for(int i=0;i<size;i++)

arr[i]=sc.nextInt();

System.out.println("before sorting:");

printArray(arr);

selectionSort(arr);

System.out.println("after soring :");

printArray(arr);

System.out.println("enter the key element to be searched:");

int key=sc.nextInt();

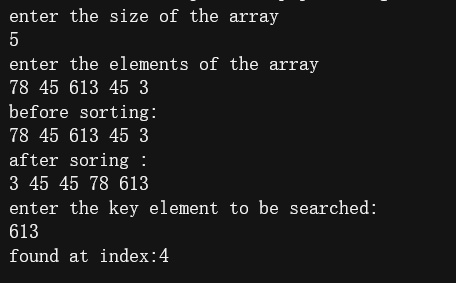
BinSearch(arr, key);

sc.close();

}

}

**Output:**

****

11. **Write a Java program that uses functions to perform the following:**

**i)sorts a list of names in ascending order using bubble sort.**

**ii)then searches for a key value(name) non recursively in the above sorted list using binary search.**

**Code:**

import java.util.Scanner;

public class NameListSort {

static void bSortNames(String[] s){

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

for(int j=0;j<s.length-i-1;j++){

if(s[j].compareTo(s[j+1])>0){

String temp=s[j];

s[j]=s[j+1];

s[j+1]=temp;

}

}

}

}

static void bSearch(String []s,String key){

boolean found = false;

int min=0,max=s.length-1;

while(min<=max&&found==false){

int mid=(min+max)/2;

if(key.equalsIgnoreCase(s[mid])){

found=true;

System.out.println("\'"+key+"\' found at index: "+mid);

break;

}

if(key.compareToIgnoreCase(s[mid])<0){

max=mid-1;

}else if(key.compareToIgnoreCase(s[mid])>0){

min=mid+1;

}

}

if(found==false)

System.out.println("\'"+key+"\' not found in the list of names.");

}

static void printArray(String[] s){

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

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

System.out.println();

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("enter the number of names you want in the list: ");

int n =sc.nextInt();

String[] names = new String[n];

System.out.println("enter the names to be added in the list");

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

names[i]=sc.next();

System.out.println("names entered are: ");

printArray(names);

bSortNames(names);

System.out.println("after sorting");

printArray(names);

System.out.println("enter the name to be searched in the list ");

String key = sc.next();

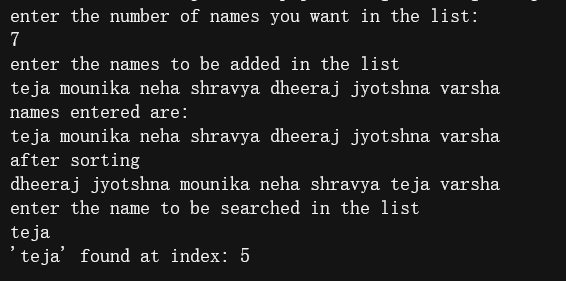
bSearch(names, key);

sc.close();

}

}

**Output:**

****

**12 .Write a Java program that sorts a list of integers passed from the command-**

**line in ascendingorder using bubble sort.**

**Code:**

public class CmdSort{

static void bSort(int[] a){

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

for(int j=0;j<a.length-1-i;j++){

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

int temp =a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

}

static void printArray(int[] a){

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

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

}

System.out.println();

}

public static void main(String [] args){

int[] arr = new int[5];

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

arr[i]=Integer.parseInt(args[i]);

}

System.out.println("before sorting:");

printArray(arr);

bSort(arr);

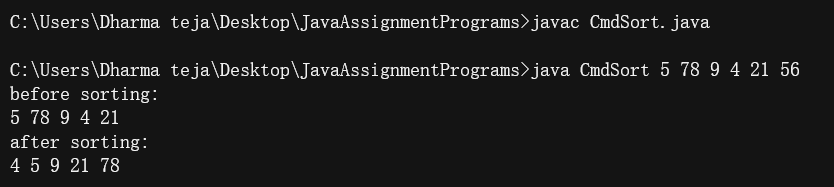
System.out.println("after sorting:");

printArray(arr);

}

}

**Output:**

****

**13. Write a Java program that uses both recursive and non recursive functions to check whether a given string is a palindrome or not. Ex: MADAM is a palindrome.**

**Code:**

import java.util.Scanner;

public class Palindrome {

static boolean isPalindrome(String s){

int l=0,r=s.length()-1;

boolean flag =true;

while(l<=r){

if(flag==true&&s.charAt(l)==s.charAt(r)){

flag=true;

}else{

flag=false;

}

l++;

r--;

}

return flag;

}

static boolean recPalindrome(String s,int left,int right){

if(s.charAt(left)!=s.charAt(right)){

return false;

}

if (left<right){

recPalindrome(s, left+1, right-1);

}

return true;

}

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

System.out.println("enter a string");

String s = new String(sc.nextLine());

if(recPalindrome(s,0,s.length()-1)){

System.out.println("it's a Palindrome!!");

}else{

System.out.println("not a palindrome");

}

sc.close();

}

}

**Output:**



**14.Write a Java program that prompts the user to enter a string and counts the number of occurrences of each letter in the string.**

**Code:**

import java.util.Scanner;

public class LetterCount {

static void letterCount(String s){

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

int count =0;

for(int j=0;j<s.length();j++){

if(s.charAt(i)==s.charAt(j)){

count++;

}

}

System.out.println("count of \'"+s.charAt(i)+"\' :"+count);

}

}

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

System.out.println("enter a string:");

String s = sc.nextLine();

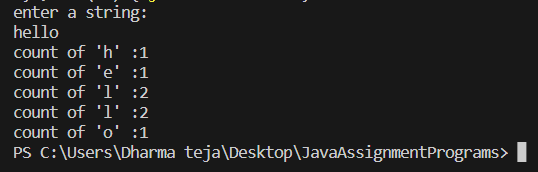
letterCount(s);

sc.close();

}

}

**Output:**



**15. Write a Java program to make frequency count of words in a given text.**

**Code:**

import java.util.Scanner;

import java.util.StringTokenizer;

public class FrequencyOFWords {

static void frequency(String []s){

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

int count=0;

for(int j=0;j<s.length;j++){

if(s[i].equalsIgnoreCase(s[j])){

count++;

}

}

System.out.println("count of \'"+s[i]+"\' :"+count);

}

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.println("enter a string: ");

String s =new String();

s=sc.nextLine();

StringTokenizer st = new StringTokenizer(s);

String[] words =new String[st.countTokens()];

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

words[i]=st.nextToken();

}

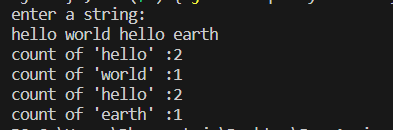
frequency(words);

sc.close();

}

}

**Output:**

****

**16.**

**a) Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.**

**Code:**

import java.io.\*;

import java.nio.file.Files;

import java.util.Scanner;

public class FilesA {

public static void main(String[] args)throws Exception {

Scanner sc = new Scanner(System.in);

System.out.println("enter a file name");

String name=sc.next();

File f=new File(name);

if(f.exists()){

System.out.println("File exists.");

if(f.canRead()){

System.out.println("file is readable!!");

}else{

System.out.println("File is not readable");

}

if(f.canWrite()){

System.out.println("file is Writable");

}else System.out.println("file is not Writable!!");

System.out.println("File type: "+Files.probeContentType(f.toPath()));

System.out.println("length of the file : "+f.length()+"bytes");

}else{

System.out.println("File dosen't exist!!");

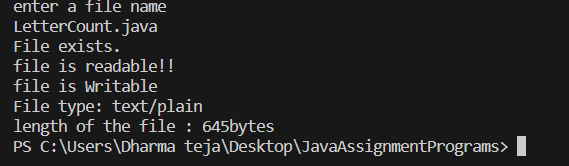
}

sc.close();

}

}

**Output:**

****

**b) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.**

**Code:**

import java.io.\*;

import java.util.Scanner;

public class FileB {

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("enter the name of the file: ");

File f1=null;

Scanner fr=null;

try{

f1=new File(sc.next());

fr=new Scanner(f1);

int i=1;

System.out.println(":::::::::::CONTENTS OF THE FILE:::::::::::::");

while(fr.hasNextLine()){

System.out.println(i+"."+fr.nextLine());

i++;

}

}catch(FileNotFoundException fnf){

System.out.println("File not Found!!!");

fnf.printStackTrace();

}catch(IOException io){

System.out.println("An Unknown IO error occured");

io.printStackTrace();

}catch(Exception e){

System.out.println("Unknown Error Occured!!!");

e.printStackTrace();

}

finally{

if(fr!=null){

fr.close();

}

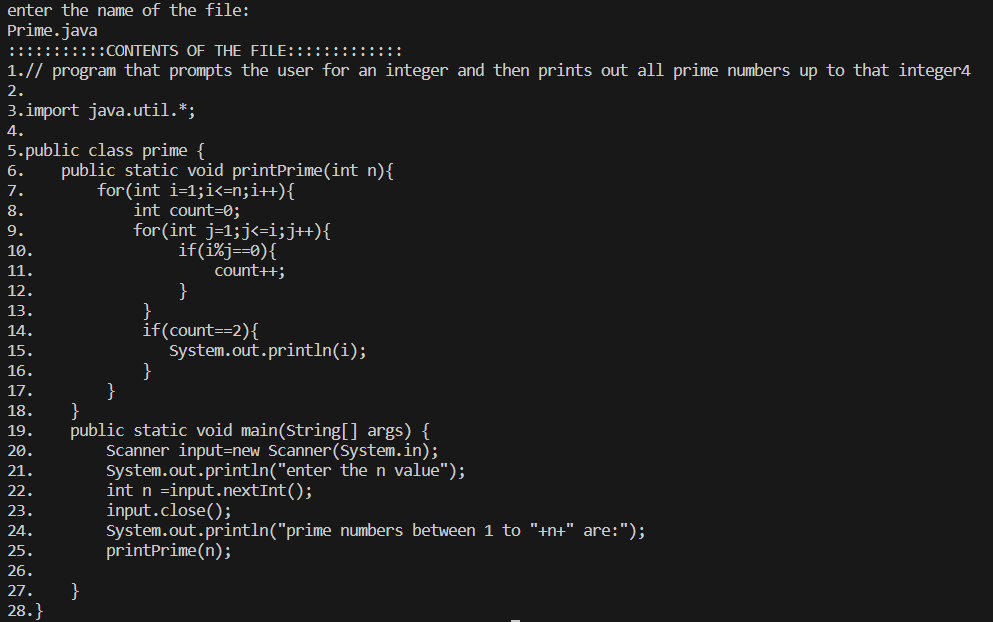
sc.close();

}

}

}

**Output:**

****

**c) Write a Java program that displays the number of characters, lines words in a text file.(FileC.java)**

**Code:**

import java.io.\*;

import java.util.Scanner;

public class FileC{

public static void main(String[] args) {

FileReader fr= null;

Scanner sc=null;

try{

fr=new FileReader("Fibonacci.java");

int count=0;

while(fr.read()!=-1){

count++;

}

fr.close();

System.out.println("count of characters: "+count);

fr=new FileReader("Fibonacci.java");

sc= new Scanner(fr);

count=0;

while(sc.hasNext()){

count++;

sc.next();

}

System.out.println("count of words: "+count);

sc.close();

fr.close();

fr=new FileReader("fibonacci.java");

sc=new Scanner(fr);

count =0;

while(sc.hasNextLine()){

count++;

sc.nextLine();

}

System.out.println("count of lines: "+count);

} catch (IOException e) {

System.out.println("Error occured");

e.printStackTrace();

}

finally{

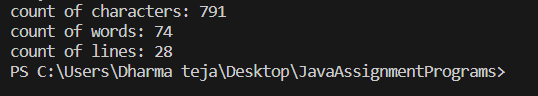
if(sc!=null) sc.close();

}

}

}

**Output:**



**d) Write a Java program to change a specific character in a file.**

**Note: Filename , number of the byte in the file to be changed and the new character are specified on the command line.**

**Code:**

import java.io.\*;

public class FileD {

public static void main(String[] args) {

File f=new File(args[0]);

try{

FileReader fr= new FileReader(f);

StringBuffer sb=new StringBuffer();

int k,pos=Integer.parseInt(args[1]),count=0;

while((k=fr.read())!=-1){

if (count==pos){

sb.append(args[2].charAt(0));

}else{

sb.append((char)k);

}

count++;

}

FileWriter pw=new FileWriter(f);

System.out.println("after changing the character at "+args[1]+" "+sb);

pw.write(sb.toString());

pw.flush();

fr.close();pw.close();

}catch(IOException io){

System.out.println("some error occured");

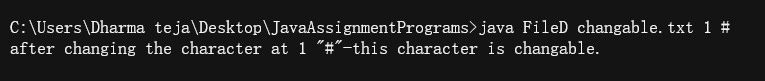
io.printStackTrace();

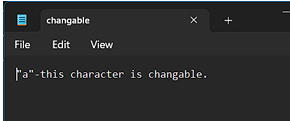
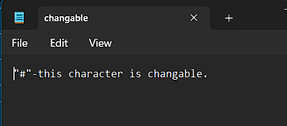
}

}

}

**Output:**

****

** **

**e.) Write a Java program that copies the contents of one file to another. File names are passed as command-line arguments. Program should handle the exceptions that may arise during the file operations.**

**Code:**

import java.io.\*;

import java.util.Scanner;

public class FileE {

public static void main(String[] args) {

File f1=null,f2=null;

Scanner sc=null;

try{

f1=new File(args[0]);

f2=new File(args[1]);

sc=new Scanner(f1);

StringBuffer buffer = new StringBuffer();

while(sc.hasNextLine()){

buffer.append("\n"+sc.nextLine());

}

PrintWriter pw = new PrintWriter(f2);

pw.println(buffer);

pw.flush();

pw.close();

sc.close();

System.out.println("copied contents of "+f1.getName()+" to "+f2.getName());

}catch(FileNotFoundException fnf){

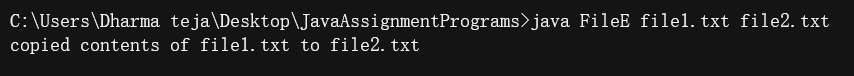
System.out.println("file Dosen't exist!!");

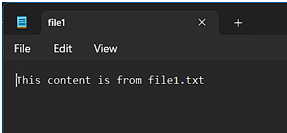
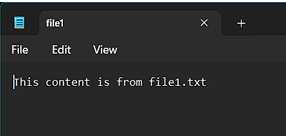
}

}

}

**Output:**

****

** **

**f) Write a Java program that reads names/numbers from a file, sorts them in ascending order,and writes them to another file.**

**Code:**

import java.io.\*;

import java.util.\*;

public class FileF{

static void Bsort(ArrayList<Integer> a){

for(int i=0;i<a.size();i++){

for(int j=0;j<a.size()-i-1;j++){

if(a.get(j)>a.get(j+1)){

int temp=a.get(j);

a.set(j,a.get(j+1));

a.set(j+1,temp);

}

}

}

}

public static void main(String[] args) {

File f1=null,f2=null;

try{

f1=new File(args[0]);

f2=new File(args[1]);

Scanner sc =new Scanner(f1);

ArrayList<Integer> list=new ArrayList<>();

while(sc.hasNextInt()){

list.add(sc.nextInt());

}

Bsort(list);

PrintWriter pw = new PrintWriter(f2);

int i=0;

while(i<list.size()){

pw.print(list.get(i)+" ");

i++;

}

pw.flush();

pw.close();

sc.close();

}catch(IOException io){

System.out.println("an error occured!!");

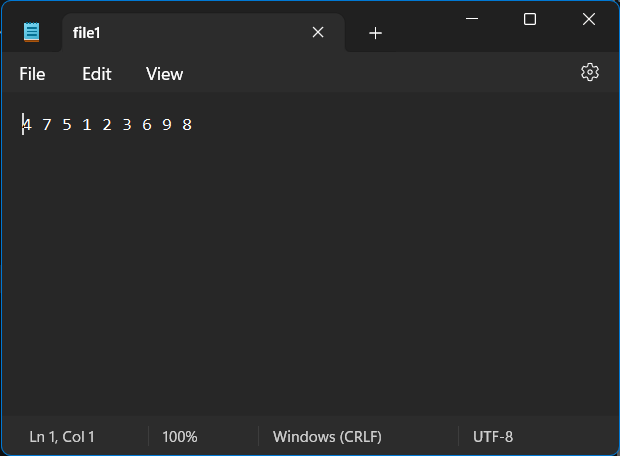
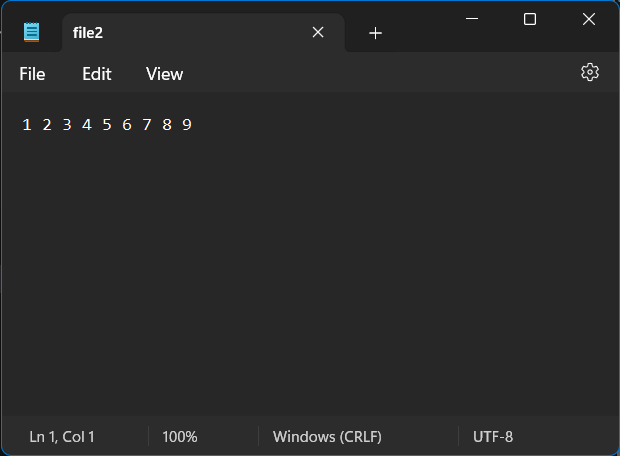
io.printStackTrace();

}

}

}

**Output:**

**17. Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers. (Use StringTokenizer class of java.util and its methods) (StringOfIntegers.java)**

**Code:**

import java.util.Scanner;

import java.util.StringTokenizer;

public class StringOfIntegers {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String numbers = new String();

System.out.println("enter the string of numbers: ");

numbers=sc.nextLine();

StringTokenizer st= new StringTokenizer(numbers);

int[] arr=new int[st.countTokens()];

int i=0;

while(st.hasMoreTokens()){

arr[i]=Integer.parseInt(st.nextToken());

i++;

}

int sum=0;

System.out.println("entered numbers: ");

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

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

sum+=arr[i];

}

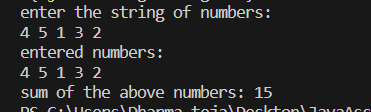
System.out.println();

System.out.println("sum of the above numbers: "+sum);

}

}

**Output:**

****

**18. Write a class in Java to represent a bank account. The class contains name of the account holder, account number, and balance amount as the data members. The class contains the member functions that allow us to have a starting balance, make deposits, make withdrawals, and get the current balance. Write appropriate methods.If insufficient funds are available, the program should raise an exception. Use your own exception class(custom exception class) and handle the exception. (BankAccount.java)**

**Code:**

import java.util.Scanner;

class InsufficientFundsException extends Exception{

}

public class BankAccount {

private double balance;

String name;

private int AcNo;

BankAccount(String s,int n){

this.name=s;

this.AcNo=n;

}

void deposit(){

Scanner sc = new Scanner(System.in);

System.out.println("enter the amount you want to deposit: ");

double amount=sc.nextDouble();

balance+=amount;

}

void withdraw()throws InsufficientFundsException{

Scanner sc = new Scanner(System.in);

System.out.println("enter the amount to withdraw: ");

double amount=sc.nextDouble();

if(balance<amount){

throw new InsufficientFundsException();

}else{

balance-=amount;

}

}

double getBalance(){

return balance;

}

public static void main(String[] args) {

BankAccount Ac1 = new BankAccount("John Cena", 001);

Scanner sc= new Scanner(System.in);

int choice;

do{

System.out.println(":::enter operarion to be performed:::\n\t1.check balance\n\t2.WithDraw\n\t3.Deposit\n\t4.exit");

choice=sc.nextInt();

switch(choice){

case 1:

System.out.println("Balance in your account : "+Ac1.getBalance());

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

break;

case 2:

try{

Ac1.withdraw();

}

catch(InsufficientFundsException I){

System.out.println("INSUFFICIENT FUNDS!!!\nenter lower amount. ");

I.printStackTrace();

}finally{

System.out.println("Current balance : "+ Ac1.getBalance());

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

}

break;

case 3:

Ac1.deposit();

System.out.println("Balance after depositing: "+Ac1.getBalance());

break;

case 4:

sc.close();

System.exit(0);

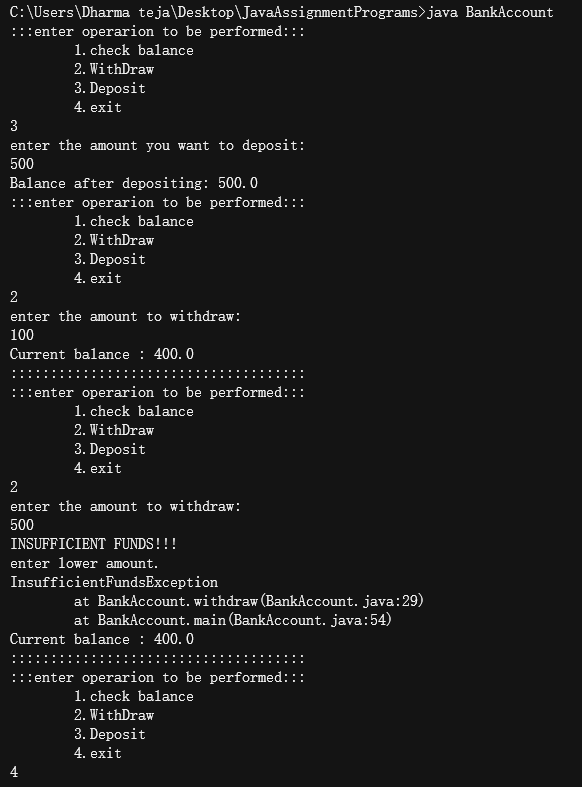
}

}while(true);

}

}

**Output:**

****

**19. Write a Java program that has an abstract class Polygon with two abstract methods area() and perimeter(). Derive three classes Rectangle, Square, and Triangle from Polygon class.Write methods to get the details of their dimensions and hence calculate their areas and perimeters The sub classes should override the two abstract methods (area() and perimeter()) by providing appropriate implementations.Demonstrate polymorphism. (polygon.java)**

**Code:**

import java.util.Scanner;

abstract class Polygon{

abstract double area();

abstract double perimeter();

}

class Square extends Polygon{

double side;

Square(){

side=0.0;

}

Square(double a){

side=a;

}

double area(){

return side\*side;

}

double perimeter(){

return 4\*side;

}

}

class Rectangle extends Polygon{

double length,width;

Rectangle(){

length=0.0;width=0.0;

}

Rectangle(double a,double b){

length=a;width=b;

}

double area(){

return length\*width;

}

double perimeter(){

return 2\*(length+width);

}

}

class Triangle extends Polygon{

double a,b,c;

Triangle(){

a=b=c=0.0;

}

Triangle(double x,double y,double z){

a=x;b=y;c=z;

}

double area(){

double s=(a+b+c)/2;

return Math.sqrt(s\*(s-a)\*(s-b)\*(s-c));

}

double perimeter(){

return a+b+c;

}

}

public class Polymorphism {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

do{

System.out.println("enter choice:\n1.Square\n2.Rectangle\n3.Triangle\n4.Exit");

int choice=sc.nextInt();

switch(choice){

case 1:

System.out.println("enter the side of the square");

double s=sc.nextDouble();

Square sq = new Square(s);

System.out.println("perimeter of the Square: "+sq.perimeter());

System.out.println("area of the square :"+sq.area());

break;

case 2:

System.out.println("enter the length and width of the Rectangle");

double l=sc.nextDouble(),w=sc.nextDouble();

Rectangle r = new Rectangle(l, w);

System.out.println("perimeter of the Rectangle: "+r.perimeter());

System.out.println("area of the Rectangle :"+r.area());

break;

case 3:

System.out.println("enter the sides of the triangle: ");

double a=sc.nextDouble(),b=sc.nextDouble(),c=sc.nextDouble();

Triangle t = new Triangle(a,b,c);

System.out.println("area of the Triangle :"+t.area());

System.out.println("perimeter of the Triangle: "+t.perimeter());

break;

case 4:

System.out.println("TERMINATED...");

sc.close();

System.exit(0);

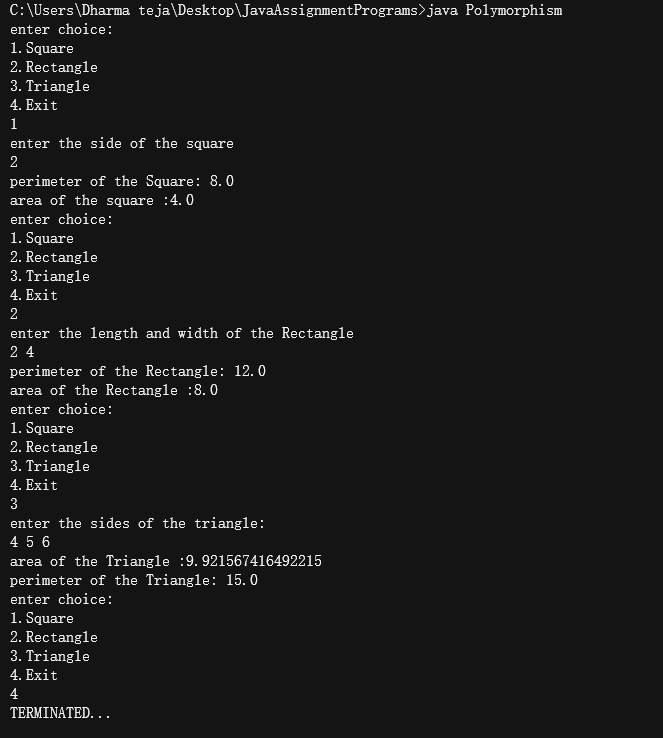
}

}while(true);

}

}

**Output:**

****

**20. Write a class for stack in Java. The class should have the methods, push(),pop(),isEmpty(),isFull() and size().Use an array representation for the stack. Write a Java program that reads a list of integers, uses the above stack methods and displays list of integers in reverse order.**

**Code:**

import java.util.Scanner;

public class Stack {

int size;

int top=-1;

int[] arr ;

Stack(int s){

size=s;

arr=new int[size];

}

boolean isEmpty(){

if(this.top==-1)

return true;

else

return false;

}

boolean isFull(){

if(this.top>=this.size){

return true;

}

else

return false;

}

void push(int a){

++top;

if(isFull()){

System.out.println("Stack overflow!!-Cannot push the element.");

top--;

return;

}

else{

arr[top]=a;

System.out.println("pushed '" +a+"' into Stack");

}

}

void pop(){

if(this.isEmpty()){

System.out.println("stack is empty!!");

}else{

System.out.println("poped '" +arr[top]+"' from Stack");

--top;

}

}

void displayStack(){

if(isEmpty()){

System.out.println("No elements in the stack!");

}else{

System.out.println("elements in the stack: ");

for(int i=top;i>=0;i--){

System.out.println("\t"+arr[i]);

}

}

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.println("enter the size of the Stack: ");

int S=sc.nextInt(),choice;

Stack s=new Stack(S);

do{

System.out.println(":::enter operarion to be performed:::\n\t1.push\n\t2.pop\n\t3.Display\n\t4.exit");

choice=sc.nextInt();

switch(choice){

case 1:

System.out.println("enter the integer to be pushed: ");

s.push(sc.nextInt());

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

break;

case 2:

s.pop();

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

break;

case 3:

s.displayStack();

break;

case 4:

System.exit(0);

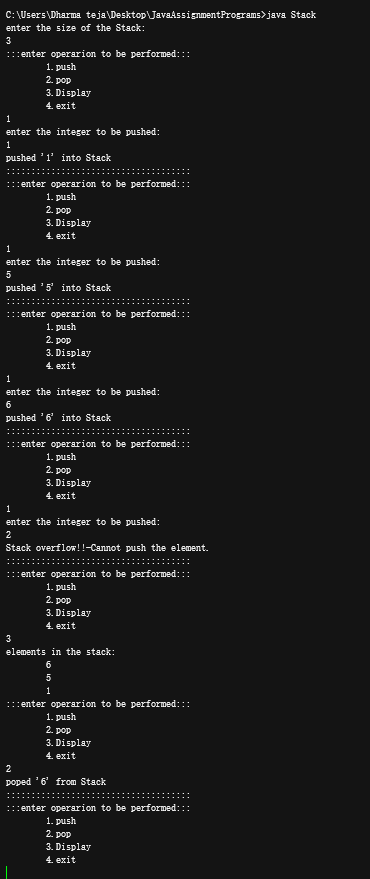
}

}while(true);

}

}

**Output:**

****

**21. Write Java program that uses functions to perform the following operations:**

**a)create a singly linked list of integers.**

**b)traverse the above list and display its contents.**

**c)delete an integer from the above list and display the contents of the list after deletion.**

**code:**

**i**mport java.util.Scanner;

public class LinkedListt{

Node head;

class Node{

int data;

Node link;

Node(int a){

this.data=a;

this.link=null;

}

}

LinkedListt(){

head=null;

}

void insert(int data){

Node newNode=new Node(data);

if(head==null){

head=newNode;

}

else{

Node temp=head;

while(temp.link!=null){

temp=temp.link;

}

temp.link=newNode;

System.out.println("inserted "+data+" into the list");

}

}

boolean delete(int key){

Node temp=head,prev=null;

boolean flag=false;

while(temp.data!=key&&temp.link!=null){

prev=temp;

temp=temp.link;

}

if(head.data==key){

head=temp.link;

System.out.println("Deleted "+key);

flag= true;

}else if(temp.data==key){

prev.link=temp.link;

System.out.println("Deleted "+key);

flag =true;

}else if(temp.data!=key&&temp.link==null){

System.out.println("No such element in the list");

}

return flag;

}

void getList(){

Node temp=head;

System.out.println("elements in the list: ");

while(temp!=null){

System.out.print(temp.data+" ");

temp=temp.link;

}

System.out.println();

}

public static void main(String[] args) {

LinkedListt list=new LinkedListt();

Scanner sc = new Scanner(System.in);

int choice;

do{

System.out.print("\n1.insert element\n2.show elements\n3.delete elements\n4.Exit\nENTER CHOICE ");

choice=sc.nextInt();

switch(choice){

case 1:

System.out.println("enter the integer to be inserted: ");

list.insert(sc.nextInt());

break;

case 2:

list.getList();

break;

case 3:

System.out.println("Enter the element to be deleted: ");

if(list.delete(sc.nextInt())){

System.out.println("after deletion: ");

list.getList();

}

break;

case 4:

sc.close();

System.exit(0);

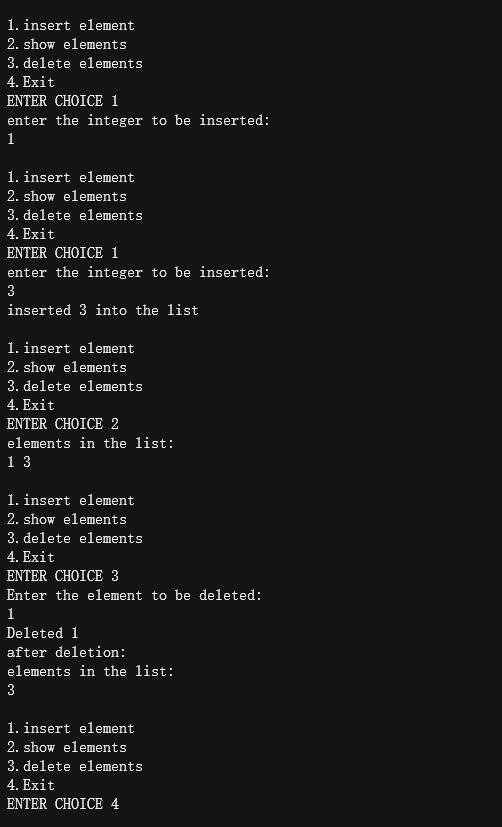
}

}while(true);

}

}

**Output:**

****