//1. Write a Java program to display all the alphabets between ‘A’ to ‘Z’ after every 2  
//seconds.  
  
 class Alphabet extends Thread  
 {  
 public void run()  
 {  
 try  
 {  
 System.out.println("First Method:");  
 for(int i=65; i<=90; i++)  
 {  
 System.out.printf("%c\t",i);  
 Thread.sleep(100);  
 }  
 //Or  
 System.out.println("\nSecond Method:\n");  
 char ch;  
 for(ch = 'A'; ch<='Z'; ch++)  
 {  
 System.out.print(ch+" ");  
 Thread.sleep(10);  
 }  
 }  
 catch(Exception e)  
 {  
 System.out.println(e);  
 }  
 }  
 }  
  
public class Slip1A {  
 public static void main(String[] args) {  
 Alphabet t = new Alphabet();  
 t.start();  
  
 }  
}  
  
  
/\*  
OUTPUT:  
First Method:  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
Second Method:  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
  
 \*/

// 2. Write a Java program to accept the details of Employee (Eno, EName, Designation,  
//Salary) from a user and store it into the database. (Use Swing)  
  
  
import java.awt.\*;  
 import javax.swing.\*;  
 import java.awt.event.\*;  
 import java.sql.\*;  
public class Slip1B extends Frame implements ActionListener  
{  
 Label l1,l2,l3,l4;  
 TextField t1,t2,t3,t4;  
 Button b;  
 Connection cn;  
 Statement st;  
 ResultSet rs;  
 public Slip1B()  
 {  
 setLayout(null);  
 l1=new Label("Eno");  
 l2=new Label("Ename");  
 l3=new Label("Salary");  
 l4=new Label("Edesignation");  
 t1=new TextField();  
 t2=new TextField();  
 t3=new TextField();  
 t4=new TextField();  
  
 b=new Button("Save");  
 l1.setBounds(50,50,100,30);  
 t1.setBounds(160,50,100,30);  
 l2.setBounds(50,90,100,30);  
 t2.setBounds(160,90,100,30);  
 l3.setBounds(50,130,100,30);  
 t3.setBounds(160,130,100,30);  
 l4.setBounds(50,170,100,30);  
 t4.setBounds(160,170,100,30);  
  
 b.setBounds(50,210,100,30);  
 add(l1);  
 add(t1);  
 add(l2);  
 add(t2);  
 add(l3);  
 add(t3);  
 add(l4);  
 add(t4);  
 add(b);  
  
 b.addActionListener(this);  
 setSize(500,500);  
 setVisible(true);  
 addWindowListener(new WindowAdapter()  
 {  
 public void windowClosing(WindowEvent e)  
 {  
 System.exit(0);  
 }  
 });  
 }  
 public void actionPerformed(ActionEvent oe)  
 {  
 String str=oe.getActionCommand();  
 if(str.equals("Save"))  
 {  
 try  
 {  
 Class.forName("com.mysql.cj.jdbc.Driver");  
 cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/java\_jdbc","root","");  
 st =cn.createStatement();  
  
 int eno = Integer.parseInt(t1.getText());  
 String ename = t2.getText();  
 int esalary = Integer.parseInt(t3.getText());  
 String edes = t4.getText();  
  
 String strr= "insert into emp values(" + eno + " ,'" + ename + "'," + esalary + ",'" + edes +"')";  
 int k=st.executeUpdate(strr);  
 if(k>0)  
 {  
  
 JOptionPane.showMessageDialog(null,"Record Is Added");  
 }  
 }  
 catch(Exception er)  
 {  
 System.out.println(er);  
 }  
 }  
 }  
 public static void main(String args[])  
 {  
 Slip1B sb = new Slip1B();  
  
 }  
}  
  
/\*  
OUTPUT:  
MariaDB [java\_jdbc]> select \* from emp;  
+-----+-------+---------+---------+  
| eno | ename | esalary | edes |  
+-----+-------+---------+---------+  
| 1 | nil | 100 | Hr |  
| 2 | omu | 1 | Manager |  
| 3 | om | 199 | boss |  
+-----+-------+---------+---------+  
3 rows in set (0.001 sec)  
 \*/

//1. Write a java program to read ‘N’ names of your friends, store it into HashSet and  
//display them in ascending order.  
  
import java.util.\*;  
  
public class Slip2A {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
 System.out.println("How many names u want to enter:");  
 int n = sc.nextInt();  
  
 HashSet hs = new HashSet(n);  
 System.out.println("Enter the Friends names:");  
 for(int i=0; i<n; i++)  
 {  
 String str = sc.next();  
 hs.add(str);  
 }  
 System.out.println("using HashSet before sorting : "+hs);  
 TreeSet tr = new TreeSet(hs);  
 System.out.println("using Treeset After Natural Sorting : "+tr);  
 }  
}  
  
/\*  
OUTPUT:  
How many names u want to enter:  
5  
Enter the Friends names:  
swapnil  
nil  
om  
omu  
avi  
  
using HashSet before sorting : [nil, avi, swapnil, om, omu]  
using Treeset After Natural Sorting : [avi, nil, om, omu, swapnil]  
  
 \*/

//Write a Java program to create LinkedList of String objects and perform the following:  
//i. Add element at the end of the list  
//ii. Delete first element of the list  
//iii. Display the contents of list in reverse order  
  
import java.util.\*;  
  
public class Slip3B {  
 public static void main(String[] args) {  
 LinkedList l = new LinkedList();  
 l.add("swapnil");  
 l.add("omu");  
 l.add("avi");  
 l.addLast("suraj");  
 System.out.println("After adding Last element :"+l);  
 l.removeFirst();  
 System.out.println("After removing first element : "+l);  
 Collections.reverse(l);  
 System.out.println("After reversing linkedlist : "+l);  
 }  
}  
  
/\*  
OUTPUT:  
After adding Last element :[swapnil, omu, avi, suraj]  
After removing first element : [omu, avi, suraj]  
After reversing linkedlist : [suraj, avi, omu]  
 \*/

//write a Java program using Runnable interface to blink Text on the frame.  
  
import java.awt.\*;  
import java.awt.event.\*;  
  
public class Slip4A extends Frame implements Runnable  
{  
 Thread t;  
 Label l1;  
 int f;  
 Slip4A()  
 {  
 t=new Thread(this);  
 t.start();  
 setLayout(null);  
 l1=new Label(" Learnwithnil ");  
 l1.setBounds(100,100,100,40);  
 add(l1);  
 setSize(300,300);  
 setVisible(true);  
 f=0;  
  
 addWindowListener(new WindowAdapter()  
 {  
 public void windowClosing(WindowEvent e)  
 {  
 System.exit(0);  
 }  
 });  
 }  
 public void run()  
 {  
 try  
 {  
 if(f==0)  
 {  
 t.sleep(200);  
 l1.setText("");  
 f=1;  
 }  
 if(f==1)  
 {  
 t.sleep(200);  
 l1.setText(" Learnwithnil ");  
 f=0;  
 }  
 }  
 catch(Exception e)  
 {  
 System.out.println(e);  
 }  
 run();  
 }  
 public static void main(String a[])  
 {  
 new Slip4A();  
 }  
}

//Write a Java program to store city names and their STD codes using an appropriate  
//collection and perform following operations:  
//i. Add a new city and its code (No duplicates)  
//ii. Remove a city from the collection  
//iii. Search for a city name and display the code  
  
  
  
 import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
import java.util.\*;  
  
  
public class Slip4B extends JFrame implements ActionListener  
 {  
 JTextField t1,t2,t3;  
 JButton b1,b2,b3;  
 JTextArea t;  
 JPanel p1,p2;  
  
 Hashtable ts;  
 Slip4B()  
 {  
 ts=new Hashtable();  
 t1=new JTextField(10);  
 t2=new JTextField(10);  
 t3=new JTextField(10);  
  
 b1=new JButton("Add");  
 b2=new JButton("Search");  
 b3=new JButton("Remove");  
  
 t=new JTextArea(20,20);  
 p1=new JPanel();  
 p1.add(t);  
  
 p2= new JPanel();  
 p2.setLayout(new GridLayout(2,3));  
 p2.add(t1);  
 p2.add(t2);  
 p2.add(b1);  
 p2.add(t3);  
 p2.add(b2);  
 p2.add(b3);  
  
 add(p1);  
 add(p2);  
  
 b1.addActionListener(this);  
 b2.addActionListener(this);  
 b3.addActionListener(this);  
  
 setLayout(new FlowLayout());  
 setSize(500,500);  
 setVisible(true);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
  
  
 }  
 public void actionPerformed(ActionEvent e)  
 {  
 if(b1==e.getSource())  
 {  
 String name = t1.getText();  
 int code = Integer.parseInt(t2.getText());  
 ts.put(name,code);  
 Enumeration k=ts.keys();  
 Enumeration v=ts.elements();  
 String msg="";  
 while(k.hasMoreElements())  
 {  
 msg=msg+k.nextElement()+" = "+v.nextElement()+"\n";  
 }  
 t.setText(msg);  
 t1.setText("");  
 t2.setText("");  
 }  
 else if(b2==e.getSource())  
 {  
 String name = t3.getText();  
  
 if(ts.containsKey(name))  
 {  
 t.setText(ts.get(name).toString());  
 }  
  
 else  
 JOptionPane.showMessageDialog(null,"City not found ...");  
 }  
 else if(b3==e.getSource())  
 {  
 String name = t3.getText();  
  
 if(ts.containsKey(name))  
 {  
 ts.remove(name);  
 JOptionPane.showMessageDialog(null,"City Deleted ...");  
 }  
  
 else  
 JOptionPane.showMessageDialog(null,"City not found ...");  
 }  
 }  
 public static void main(String a[])  
 {  
 new Slip4B();  
 }  
 }

/\*7. Write a java program to accept ‘N’ integers from a user.  
 Store and display integers in sorted order having proper collection class.  
The collection should not accept duplicate elements.  
  
 \*/  
  
import java.util.Scanner;  
import java.util.TreeSet;  
  
public class SortInt7 {  
  
 public static void main(String[] args) {  
 TreeSet t = new TreeSet();  
 Scanner sc = new Scanner(System.in);  
 System.out.println("Enter the no of integers:");  
 int n = sc.nextInt();  
  
 System.out.println("Enter the numbers:");  
 for(int i=0; i<n; i++)  
 {  
 t.add(sc.nextInt());  
 }  
  
 System.out.println("By default elements are inserted according to the defalut natural soring order into Treeset:");  
 System.out.println("so the numbers are sorting order:"+t);  
 }  
}  
  
/\*OUTPUT:  
Enter the no of integers:  
7  
Enter the numbers:  
99  
1  
34  
50  
88  
77  
66  
By default elements are inserted according to the defalut natural soring order into Treeset:  
so the numbers are sorting order:[1, 34, 50, 66, 77, 88, 99]  
  
 \*/

/\*8. Write a java program to accept ‘N’ Integers from a user store them into LinkedList Collection  
 and display only negative integers.  
  
 \*/  
  
import java.util.Scanner;  
import java.util.\*;  
  
public class NegativeNo {  
  
 public static void main(String[] args) {  
 LinkedList t = new LinkedList();  
 Scanner sc = new Scanner(System.in);  
 System.out.println("Enter the no of integers:");  
 int n = sc.nextInt();  
  
 System.out.println("Enter the numbers:");  
 for(int i=0; i<n; i++)  
 {  
 t.add(sc.nextInt());  
 }  
  
 System.out.println(t);  
 ListIterator lt = t.listIterator();  
 System.out.println("The negative numbers are:");  
 while(lt.hasNext())  
 {  
 int no = (int) lt.next();  
 if(no < 0)  
 System.out.println(no);  
 }  
  
 }  
 }  
  
 /\* OUTPUT:  
 Enter the no of integers:  
5  
Enter the numbers:  
23  
-55  
45  
-3  
-99  
[23, -55, 45, -3, -99]  
The negative numbers are:  
-55  
-3  
-99  
 \*/

/\*9. Write a java program to accept ‘N’ Subject Names from a user store them into LinkedList Collection  
 and Display them by using Iterator interface.  
  
 \*/  
  
  
import java.util.Scanner;  
 import java.util.\*;  
  
public class Subject9 {  
  
 public static void main(String[] args) {  
 LinkedList t = new LinkedList();  
 Scanner sc = new Scanner(System.in);  
 System.out.println("Enter the no of subjects:");  
 int n = sc.nextInt();  
  
 System.out.println("Enter the subjects:");  
 for(int i=0; i<n; i++)  
 {  
 t.add(sc.next());  
 }  
  
 System.out.println(t);  
 Iterator it = t.iterator();  
 System.out.println("The subjects using Iterator:");  
 while(it.hasNext())  
 {  
 System.out.println(it.next());  
 }  
  
 }  
}  
/\*  
OUTPUT:  
Enter the no of subjects:  
5  
Enter the subjects:  
math  
english  
art  
AI  
sketch  
[math, english, art, AI, sketch]  
The subjects using Iterator:  
math  
english  
art  
AI  
sketch  
 \*/

/\*10. Write a java program to accept ‘N’ student names through command line, store them into the appropriate Collection  
 and display them by using Iterator and ListIterator interface.  
  
 \*/  
  
import java.util.Scanner;  
 import java.util.\*;  
  
public class CommandName {  
  
 public static void main(String[] args) throws Exception  
 {  
 LinkedList t = new LinkedList();  
 Scanner sc = new Scanner(System.in);  
  
 for(int i=0; i<args.length; i++)  
 {  
 String str = args[i];  
 t.add(str);  
 }  
  
 System.out.println(t);  
  
 Iterator it = t.iterator();  
 System.out.println("The names using Iterator:");  
 while(it.hasNext())  
 {  
 System.out.println(it.next());  
 }  
  
 ListIterator lt = t.listIterator();  
 System.out.println("The names using ListIterator:");  
 while(lt.hasNext())  
 {  
 System.out.println(lt.next());  
 }  
  
 }  
}  
  
/\*  
OUTPUT:  
 C:\Users\swapniljamble\IdeaProjects\MyProject\src> javac CommandName.java  
 C:\Users\swapniljamble\IdeaProjects\MyProject\src> java CommandName swapnil nil omu avi raju sam tej wednesday jack  
[swapnil, nil, omu, avi, raju, sam, tej, wednesday, jack]  
The names using Iterator:  
swapnil  
nil  
omu  
avi  
raju  
sam  
tej  
wednesday  
jack  
The names using ListIterator:  
swapnil  
nil  
omu  
avi  
raju  
sam  
tej  
wednesday  
jack  
  
 \*/

/\*11. Write a Java program to create LinkedList of integer objects and perform the following:  
 i. Add element at first position  
 ii. Delete last element  
 iii. Display the size of link list  
 \*/  
  
import java.util.Scanner;  
 import java.util.\*;  
  
public class LinkedListSize {  
  
 public static void main(String[] args) {  
 LinkedList l = new LinkedList();  
  
 l.add(50);  
 l.add(30);  
 l.add(20);  
 l.addFirst(88);  
 System.out.println("After adding first element in the LinkedList:" + l);  
 l.removeLast();  
 System.out.println("after removing last element from the LinkedList:" + l);  
 int size = l.size();  
 System.out.println("The size of the LinkedList:" + size);  
  
 }  
}  
  
/\*OUTPUT:  
After adding first element in the LinkedList:[88, 50, 30, 20]  
after removing last element from the LinkedList:[88, 50, 30]  
The size of the LinkedList:3  
 \*/

Multithreading :

// java program for generate random number and find prime numbers from random number using thread concept.  
import java.util.\*;  
  
class PrimeNum extends Thread  
{  
 int n;  
 PrimeNum(int n)  
 {  
 this.n = n;  
 }  
 public void run()  
 {  
  
 int i,num,count;  
  
 System.out.println("prime number between 1 to "+n);  
 for(num=1;num<=n;num++)  
 {  
 count=0;  
 for(i=2;i<=num/2; i++)  
 {  
 if(num%i==0)  
 {  
 count++;  
 break;  
 }  
 }  
 if(count==0 && num!=1)  
 System.out.println(num);  
 }  
 }  
}  
class RandNum extends Thread  
{  
 Random random = new Random();  
 public void run() {  
 for (int i = 0; i < 5; i++) {  
 int num = random.nextInt(50);  
 System.out.println("random number is:"+num);  
  
 PrimeNum pn = new PrimeNum(num);  
 pn.start();  
 try {  
 pn.join();  
 }  
 catch(Exception e){}  
 }  
 }  
}  
public class RandomPrimeNum {  
 public static void main(String[] args) {  
 RandNum rn = new RandNum();  
 rn.start();  
 }  
}  
  
  
/\* OUTPUT:  
random number is:37  
prime number between 1 to 37  
2  
3  
5  
7  
11  
13  
17  
19  
23  
29  
31  
37  
random number is:3  
prime number between 1 to 3  
2  
3  
random number is:15  
prime number between 1 to 15  
2  
3  
5  
7  
11  
13  
random number is:34  
prime number between 1 to 34  
2  
3  
5  
7  
11  
13  
17  
19  
23  
29  
31  
random number is:10  
prime number between 1 to 10  
2  
3  
5  
7  
 \*/

/\*Write a java program to accept a String from a user and display each vowel from a   
 String after every 3 seconds.   
 \*/  
  
import java.util.Scanner;  
  
public class Vowel3Sec {  
 public static void main(String[] args) throws Exception{  
 Scanner sc = new Scanner(System.in);  
 System.out.println("Enter the String:");  
 String str = sc.nextLine();  
  
 for(int i=0; i<str.length(); i++)  
 {  
 if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o' || str.charAt(i) == 'u') {  
 System.out.println(str.charAt(i));  
 Thread.sleep(3000);  
 }  
 }  
 }  
}

import java.util.\*;  
  
class prodcons {  
 LinkedList list = new LinkedList();  
  
 public void produce() throws InterruptedException  
 {  
 int value = 0;  
 while (true)  
 {  
 synchronized (this)  
 {  
 while (list.size() > 0)  
 wait();  
  
 System.out.println("Produced data :" + value);  
 list.add(value);  
 value++;  
 notify();  
  
 Thread.sleep(1000);  
 }  
 }  
 }  
 public void consume() throws InterruptedException  
 {  
 while (true)  
 {  
 synchronized (this)  
 {  
 while (list.size() == 0)  
 wait();  
  
 int value = (int) list.removeFirst();  
 System.out.println("Consume data :" + value);  
 notify();  
  
 Thread.sleep(1000);  
  
  
 }  
 }  
 }  
}  
  
class PC {  
 public static void main(String[] args) throws Exception {  
 prodcons pc = new prodcons();  
 Thread t1 = new Thread(new Runnable() {  
 public void run()  
 {  
 try {  
 pc.produce();  
 }  
 catch (Exception e) { }  
 }  
 });  
  
 Thread t2 = new Thread(new Runnable() {  
 @Override  
 public void run() {  
 try {  
 pc.consume();  
 }  
 catch (Exception e) {}  
 }  
 });  
  
 t1.start();  
 t2.start();  
  
 t1.join();  
 t2.join();  
 }  
}  
  
  
/\*OUTPUT:  
ProducerConsumer  
Produced data :0  
Consume data :0  
Produced data :1  
Consume data :1  
Produced data :2  
Consume data :2  
Produced data :3  
Consume data :3  
Produced data :4  
Consume data :4  
Produced data :5  
Consume data :5  
.  
.  
 \*/

import java.util.HashMap;  
import java.util.TreeMap;  
  
public class HashmapSort {  
 public static void main(String args[])  
 {  
 HashMap map = new HashMap();  
  
 // putting values in the Map  
 map.put("Jayant", 80);  
 map.put("Abhishek", 90);  
 map.put("Anushka", 80);  
 map.put("Amit", 75);  
 map.put("Danish", 40);  
  
 System.out.println(map);  
 TreeMap tree = new TreeMap(map);  
 System.out.println(tree);  
 }  
}

/\*  
rogram to define a thread for printing text on output screen for „n‟ number of  
times.Create 3 threads and run them. Pass the text „n‟ parameters to the thread  
constructor. Example:  
i. First thread prints “I AM In FY” 10 times.  
 ii. Second thread prints “I AM In SY” 20 times  
iii. Third thread prints “I AM In TY” 30 times  
 \*/  
  
class thread111 extends Thread  
{  
 public void run()  
 {  
 for(int i=0; i<1; i++)  
 {  
 System.out.println("I am in fy");  
 System.out.println("name"+Thread.currentThread().getName());  
 System.out.println("priority:"+Thread.currentThread().getPriority());  
 }  
 }  
}  
public class Threads3 {  
 public static void main(String[] args) {  
  
thread111 t1 = new thread111();  
t1.start();  
 System.out.println("naem:"+Thread.currentThread().getName());  
 System.out.println("priority:"+Thread.currentThread().getPriority());  
 }  
}

//java program for generate random number and create 10 thread to sum and average.  
  
class ThreadSum1 extends Thread  
{  
 int start ,end;  
 int sum;  
 static int allsum;  
 ThreadSum1(int start , int end)  
 {  
 this.start = start;  
 this.end = end;  
 }  
 public void run()  
 {  
 for(int i = start; i<=end; i++)  
 {  
 sum += i;  
 }  
 allsum +=sum;  
 }  
 static int getAvg()  
 {  
 return allsum/100;  
 }  
}  
public class SumAvgThread {  
 public static void main(String[] args) throws InterruptedException {  
  
 int start = 0, end=99;  
 ThreadSum1 ts[] = new ThreadSum1[10];  
  
 for(int i=0; i<10; i++)  
 {  
 ts[i] = new ThreadSum1(++start,start+=end);  
 ts[i].start();  
 ts[i].join();  
 }  
  
 System.out.println("Sum of number between 1 to 1000 :"+ThreadSum1.allsum);  
 System.out.println("Average is: "+ThreadSum1.getAvg());  
 }  
}  
  
/\* OUTPUT:  
Sum of number between 1 to 1000 :500500  
Average is: 5005  
  
 \*/

/\*11) Program to define a thread for printing text on output screen for „n‟ number of times.  
 Create 3 threads and run them. Pass the text „n‟ parameters to the thread constructor. Example:  
i. First thread prints “COVID19” 10 times.  
ii. Second thread prints “LOCKDOWN2020” 20 times  
iii. Third thread prints “VACCINATED2021” 30 times  
  
 \*/  
class Thread1 extends Thread  
{  
 public void run()  
 {  
 for(int i=0; i<10; i++)  
 System.out.println("COVID19");  
 }  
}  
  
class Thread2 extends Thread  
{  
 public void run()  
 {  
 for(int i=0; i<20; i++)  
 System.out.println("LOCKDOWN2020");  
 }  
}  
class Thread3 extends Thread  
{  
 public void run()  
 {  
 for(int i=0; i<30; i++)  
 System.out.println("VACCINATED2021");  
 }  
}  
public class ThreadText {  
 public static void main(String[] args) {  
 Thread1 t1= new Thread1();  
 Thread2 t2= new Thread2();  
 Thread3 t3= new Thread3();  
 t2.start();  
 t1.start();  
 t3.start();  
  
 }  
}  
  
/\* OUTPUT:  
LOCKDOWN2020  
COVID19  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
COVID19  
COVID19  
COVID19  
COVID19  
COVID19  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
LOCKDOWN2020  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
LOCKDOWN2020  
LOCKDOWN2020  
COVID19  
COVID19  
COVID19  
COVID19  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
VACCINATED2021  
  
 \*/

/\*12) Write a program in which thread sleep for 6 sec in the loop in reverse order from 100 to 1  
and change the name of thread.  
  
 \*/  
  
 class ReversedOrder extends Thread  
 {  
 public void run()  
 {  
 for(int i=100; i>=1; i--)  
 {  
  
 System.out.println(i);  
 try {  
 Thread.sleep(6000);  
 }  
 catch (InterruptedException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
 }  
public class ThreadSleep {  
 public static void main(String[] args) throws Exception {  
 ReversedOrder rv = new ReversedOrder();  
 rv.start();  
 System.out.println("The default name of thread:"+rv.getName());  
 rv.setName("swapnil");  
 System.out.println("After changing name of Thread:"+rv.getName());  
  
 }  
}  
  
  
/\* OUTPUT:  
100  
The default name of thread:Thread-0  
After changing name of Thread:swapnil  
99  
98  
97  
96  
95  
.  
.  
.  
 \*/

import java.util.\*;  
  
 class ProducerConsumerProblem {  
 LinkedList list = new LinkedList();  
  
 public void produce() throws InterruptedException {  
 int value = 0;  
 while (true) {  
 synchronized (this) {  
 while (list.size() > 0)  
 wait();  
  
 System.out.println("Produced data :" + value);  
 list.add(value);  
 value++;  
 notify();  
  
 Thread.sleep(1000);  
 }  
 }  
 }  
 public void consume() throws InterruptedException {  
 while (true) {  
 synchronized (this) {  
 while (list.size() == 0)  
 wait();  
  
 int value = (int) list.removeFirst();  
 System.out.println("Consume data :" + value);  
 notify();  
  
 Thread.sleep(1000);  
  
  
 }  
 }  
 }  
}  
  
 class ProducerConsumer {  
 public static void main(String[] args) throws Exception {  
 ProducerConsumerProblem pc = new ProducerConsumerProblem();  
 Thread t1 = new Thread(new Runnable() {  
 public void run() {  
 try {  
 pc.produce();  
 } catch (InterruptedException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 });  
  
 Thread t2 = new Thread(new Runnable() {  
 @Override  
 public void run() {  
 try {  
 pc.consume();  
 } catch (InterruptedException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 });  
  
 t1.start();  
 t2.start();  
  
 t1.join();  
 t2.join();  
 }  
 }  
  
  
/\*OUTPUT:  
ProducerConsumer  
Produced data :0  
Consume data :0  
Produced data :1  
Consume data :1  
Produced data :2  
Consume data :2  
Produced data :3  
Consume data :3  
Produced data :4  
Consume data :4  
Produced data :5  
Consume data :5  
.  
.  
 \*/

class ThreadSum extends Thread  
{  
 int start ,end;  
 int sum;  
 ThreadSum(int start , int end)  
 {  
 this.start = start;  
 this.end = end;  
 }  
 public void run()  
 {  
 for(int i = start; i<=end; i++)  
 {  
 sum += i;  
 }  
 }  
 int getSum()  
 {  
 return sum;  
 }  
}  
  
public class add1000 {  
 public static void main(String[] args) throws InterruptedException {  
  
  
 ThreadSum t1 = new ThreadSum(1,100);  
 t1.start();  
 t1.join();  
 System.out.println(t1.getSum());  
  
 ThreadSum t2 = new ThreadSum(101,200);  
 t2.start();  
 t2.join();  
 System.out.println(t2.getSum());  
  
 ThreadSum t3 = new ThreadSum(201,300);  
 t3.start();  
 t3.join();  
 System.out.println(t3.getSum());  
  
 ThreadSum t4 = new ThreadSum(301,400);  
 t4.start();  
 t4.join();  
 System.out.println(t4.getSum());  
 ThreadSum t5 = new ThreadSum(401,500);  
 t5.start();  
 t5.join();  
 System.out.println(t5.getSum());  
  
 ThreadSum t6 = new ThreadSum(501,600);  
 t6.start();  
 t6.join();  
 System.out.println(t6.getSum());  
 ThreadSum t7 = new ThreadSum(601,700);  
 t7.start();  
 t7.join();  
 System.out.println(t7.getSum());  
  
 ThreadSum t8 = new ThreadSum(701,800);  
 t8.start();  
 t8.join();  
 System.out.println(t8.getSum());  
  
 ThreadSum t9 = new ThreadSum(801,900);  
 t9.start();  
 t9.join();  
 System.out.println(t9.getSum());  
  
 ThreadSum t10 = new ThreadSum(901,1000);  
 t10.start();  
 t10.join();  
 System.out.println(t10.getSum());  
  
 int sum=0;  
 sum = t1.getSum() +t2.getSum() +t3.getSum() +t4.getSum() +t5.getSum() +t6.getSum() +t7.getSum() +t8.getSum() +t9.getSum() +t10.getSum();  
 System.out.println("The addition of numbers between 1 to 1000 :"+sum);  
 }  
}  
  
/\* OUTPUT:  
5050  
15050  
25050  
35050  
45050  
55050  
65050  
75050  
85050  
95050  
The addition of numbers between 1 to 1000 :500500  
 \*/

/\*Write a program for a simple search engine. Accept a string to be searched. Search   
for thestring in all text files in the current folder. Use a separate thread for each file.   
The result should display the filename, line number where the string is found.\*/  
import java.io.\*;  
class FileWatcher extends Thread  
{  
 String filename;  
 int count = 0,countBuffer=0,countLine=0;  
 String lineNumber = "";  
 BufferedReader br;  
 String inputSearch;  
 String line = "";  
 FileWatcher(String filename,String srch)  
 {  
 this.filename = filename;  
 inputSearch = srch;  
 }  
 public void run()  
 {  
 try  
 {  
 try  
 {  
 br = new BufferedReader(new FileReader("/root/TextFiles/"+filename));  
 try  
 {  
 while ((line = br.readLine()) != null)  
 {  
 countLine++;  
 //System.out.println(line);  
 String[] words = line.split(" ");  
 for (String word : words)  
 {  
 if (word.equals(inputSearch))  
 {  
 count++;  
 countBuffer++;  
 }  
 }  
 if (countBuffer > 0)  
 {  
 countBuffer = 0;  
 lineNumber += countLine + ",";  
 }  
 }  
 br.close();  
 } catch (IOException e)  
 {  
 e.printStackTrace();  
 }  
 } catch (FileNotFoundException e)  
 {  
 e.printStackTrace();  
 }  
 System.out.println("\n Thread Name = "+this+"\n File Name = "+filename+"\n Times found at = "+count+"\n Word found at = "+lineNumber);  
 }  
 catch(Exception e)  
 {  
 e.printStackTrace();  
 }  
 }  
}  
class Ass2SetB2  
{  
 public static void main(String args[])  
 {  
 int i;  
 File fl;  
 File dir = new File("/root/TextFiles/");  
 String files[] = dir.list();  
 FileWatcher fw[] = new FileWatcher[files.length];  
 for(i=0; i<files.length; i++)  
 {  
 if(files[i].endsWith(".txt"))  
 {  
 fw[i] = new FileWatcher(files[i],args[0]);  
 fw[i].start();  
 }  
 }  
 }  
}

Database :

/\*1) Write a program to display Student table data (rollno, name, mobno, percentage)  
  
 \*/  
  
import java.sql.\*;  
public class Q1select {  
 public static void main(String[] args) throws Exception  
 {  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; // String jdbc\_url = "jdbc:postgresql://172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
  
 Class.forName("com.mysql.cj.jdbc.Driver"); // " Class.forName("org.postgresql.Driver";  
 Connection con = DriverManager.getConnection(jdbc\_url,user,pwd);  
 Statement st = con.createStatement();  
 String query = "select \* from student";  
  
 ResultSet rs = st.executeQuery(query);  
 while(rs.next())  
 {  
 System.out.println(rs.getInt(1)+" "+rs.getString(2)+ " "+rs.getDouble(3)+" "+rs.getInt(4));  
 }  
 con.close();  
 }  
}  
  
/\*OUTPUT:  
1 nil 88.3 343  
22 sd 33.0 0  
2 om 88.34 444  
3 nill 33.3 453  
5 raju 34.0 657  
8 dk 76.0 656  
  
 \*/

/\*22) Write a program to perform insert and delete operations  
 on employee table using PreparedStatement (Empid, Empname, Empsalary)  
  
 \*/  
import java.sql.\*;  
import java.util.Scanner;  
  
public class Q22InsetDelete {  
 public static void main(String[] args) throws Exception {  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; // String jdbc\_url = "jdbc:postgresql://172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
 String query = "insert into student (id,name,percentage,mno)values(?,?,?,?)";  
 String query2 = "delete from student where id =?";  
  
 Class.forName("com.mysql.cj.jdbc.Driver"); // " Class.forName("org.postgresql.Driver";  
  
 Scanner sc = new Scanner(System.in);  
 Connection con = DriverManager.getConnection(jdbc\_url,user,pwd);  
 PreparedStatement pst = con.prepareStatement(query);  
  
 System.out.println("Enter the Student id:");  
 int id = sc.nextInt();  
 System.out.println("Enter the Student name:");  
 String name = sc.next();  
 System.out.println("Enter the Student percentage:");  
 Double percentage = sc.nextDouble();  
 System.out.println("Enter the Student mobile no:");  
 int mno = sc.nextInt();  
  
 pst.setInt(1,id);  
 pst.setString(2,name);  
 pst.setDouble(3,percentage);  
 pst.setInt(4,mno);  
 pst.executeUpdate();  
  
 PreparedStatement pst2 = con.prepareStatement(query2);  
 System.out.println("Enter the id which row u want to delete:");  
 int id2 = sc.nextInt();  
  
 pst2.setInt(1,id2);  
 pst2.executeUpdate();  
  
  
  
  
 }  
}  
  
  
/\*  
Enter the Student id:  
5  
Enter the Student name:  
raju  
Enter the Student percentage:  
34  
Enter the Student mobile no:  
657  
MariaDB [java\_jdbc]> select \* from student;  
+------+------+------------+------+  
| id | name | percentage | mno |  
+------+------+------------+------+  
| 1 | nil | 88.3 | 343 |  
| 22 | sd | 33 | NULL |  
| 2 | om | 88.34 | 444 |  
| 3 | nill | 33.3 | 453 |  
| 5 | raju | 34 | 657 |  
| 6 | omm | 36 | 677 |  
| 6 | omm | 36 | 677 |  
+------+------+------------+------+  
7 rows in set (0.000 sec)  
  
Enter the Student id:  
8  
Enter the Student name:  
dk  
Enter the Student percentage:  
76  
Enter the Student mobile no:  
656  
Enter the id which row u want to delete:  
6  
  
MariaDB [java\_jdbc]> select \* from student;  
+------+------+------------+------+  
| id | name | percentage | mno |  
+------+------+------------+------+  
| 1 | nil | 88.3 | 343 |  
| 22 | sd | 33 | NULL |  
| 2 | om | 88.34 | 444 |  
| 3 | nill | 33.3 | 453 |  
| 5 | raju | 34 | 657 |  
| 8 | dk | 76 | 656 |  
+------+------+------------+------+  
6 rows in set (0.001 sec)  
  
Enter the Student id:  
8  
Enter the Student name:  
dk  
Enter the Student percentage:  
76  
Enter the Student mobile no:  
656  
Enter the id which row u want to delete:  
6  
  
 \*/

/\*23) Write a program to display information about the database and list all the tables in the database. (Use DatabaseMetaData).  
  
 \*/  
  
import java.sql.Connection;  
import java.sql.DatabaseMetaData;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.util.Scanner;  
  
public class Q23InfoData {  
 public static void main(String[] args) throws Exception {  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; // String jdbc\_url = "jdbc:postgresql://172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
 Class.forName("com.mysql.cj.jdbc.Driver"); // " Class.forName("org.postgresql.Driver";  
  
 Connection con = DriverManager.getConnection(jdbc\_url, user, pwd);  
 DatabaseMetaData dbms = con.getMetaData();  
 ResultSet rs = dbms.getTables(null, null, null, new String[]{"TABLE"});  
 System.out.println("List of Tables:");  
 while (rs.next()) {  
 String tblName = rs.getString("TABLE\_NAME");  
 System.out.println("Table:" + tblName);  
 }  
 con.close();  
 }  
}  
  
/\*  
List of Tables:  
Table:activity\_log  
Table:answer  
Table:assignment  
Table:class  
Table:class\_quiz  
Table:class\_subject\_overview  
Table:content  
Table:department  
Table:event  
Table:files  
Table:message  
Table:message\_sent  
Table:notification  
Table:notification\_read  
Table:notification\_read\_teacher  
Table:question\_type  
  
 \*/

import java.sql.\*;  
public class Q24Column {  
 public static void main(String args[]){  
 try{  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; // String jdbc\_url = "jdbc:postgresql://172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
 Class.forName("com.mysql.cj.jdbc.Driver"); // " Class.forName("org.postgresql.Driver";  
 Connection con = DriverManager.getConnection(jdbc\_url, user, pwd);  
  
 PreparedStatement ps=con.prepareStatement("select \* from student");  
 ResultSet rs=ps.executeQuery();  
 ResultSetMetaData rsmd=rs.getMetaData();  
  
 System.out.println("Total columns: " + rsmd.getColumnCount());  
  
 int i=1;  
 while(rs.next()) {  
  
 System.out.println("Column Name of 1st column: " + rsmd.getColumnName(i));  
 System.out.println("Column Type Name of 1st column: " + rsmd.getColumnTypeName(i));  
 i++;  
 }  
 con.close();  
 }catch(Exception e){ System.out.println(e);}  
 }  
}  
  
/\*  
Total columns: 4  
Column Name of 1st column: id  
Column Type Name of 1st column: INT  
Column Name of 1st column: name  
Column Type Name of 1st column: VARCHAR  
Column Name of 1st column: percentage  
Column Type Name of 1st column: DOUBLE  
Column Name of 1st column: mno  
Column Type Name of 1st column: INT  
java.sql.SQLException: Column index out of range.  
  
 \*/

import java.sql.\*;  
import java.util.Scanner;  
import java.util.function.Predicate;  
  
/\*25) Create a Employee table with fields empid, empname,  
empsalary. Insert values in the table. Write a menu driven program to perform the following operations on Employee table.  
1. View All 2. Search 3. Exit  
  
 \*/  
public class Q25MenuDriven {  
 public static void main(String[] args) throws Exception{  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; // String jdbc\_url = "jdbc:postgresql://172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
 Class.forName("com.mysql.cj.jdbc.Driver"); // " Class.forName("org.postgresql.Driver";  
 Connection con = DriverManager.getConnection(jdbc\_url, user, pwd);  
 Scanner sc = new Scanner(System.in);  
  
 String query = "create table employee1(eid int,ename varchar(100),esalary int)";  
  
 Statement st = con.createStatement();  
 st.executeUpdate(query);  
  
  
 int choice;  
 do {  
 System.out.println("1.view all\n 2.search \n3. Insert \n 4. Exit");  
 System.out.println("Enter ur choice:");  
 choice = sc.nextInt();  
  
 switch(choice)  
 {  
 case 1:  
 String select = "select \* from employee1";  
 Statement st1 = con.createStatement();  
 ResultSet rs = st1.executeQuery(select);  
  
 while(rs.next())  
 {  
 System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getInt(3));  
 }  
 break;  
  
 case 2 :  
 String search = "select \* from employee1 where eid =?";  
 PreparedStatement pst1 = con.prepareStatement(search);  
 System.out.println("Enter the employee id which u want to be searched:");  
 int id = sc.nextInt();  
  
 pst1.setInt(1,id);  
 ResultSet rs1 = pst1.executeQuery();  
 while(rs1.next())  
 {  
 System.out.println(rs1.getInt(1)+" "+rs1.getString(2)+" "+rs1.getInt(3));  
 }  
  
  
 break;  
  
 case 3:  
 String query2 = "insert into employee1(eid,ename,esalary)values(?,?,?)";  
 PreparedStatement pst = con.prepareStatement(query2);  
  
 System.out.println("Enter the emp id:");  
 int eid = sc.nextInt();  
 System.out.println("Enter the emp name:");  
 String ename = sc.next();  
 System.out.println("Enter the emp salary:");  
 int esalary = sc.nextInt();  
  
 pst.setInt(1,eid);  
 pst.setString(2,ename);  
 pst.setDouble(3,esalary);  
 pst.executeUpdate();  
 }  
 }while(choice!=4);  
  
 }  
}  
  
/\*  
OUTPUT:  
1.view all  
 2.search  
3. Insert  
 4. Exit  
Enter ur choice:  
3  
Enter the emp id:  
1  
Enter the emp name:  
swapnil  
Enter the emp salary:  
8999  
1.view all  
 2.search  
3. Insert  
 4. Exit  
Enter ur choice:  
3  
Enter the emp id:  
2  
Enter the emp name:  
omu  
Enter the emp salary:  
999  
1.view all  
 2.search  
3. Insert  
 4. Exit  
Enter ur choice:  
3  
Enter the emp id:  
3  
Enter the emp name:  
avi  
Enter the emp salary:  
888  
1.view all  
 2.search  
3. Insert  
 4. Exit  
Enter ur choice:  
1  
1 swapnil 8999  
2 omu 999  
3 avi 888  
1.view all  
 2.search  
3. Insert  
 4. Exit  
Enter ur choice:  
2  
Enter the employee id which u want to be searched:  
1  
1 swapnil 8999  
1.view all  
 2.search  
3. Insert  
 4. Exit  
 \*/

/\*27) Create a Project table with fields Proj\_id, Proj\_name, Proj\_status.  
Insert values in the table. Write a menu driven program to perform the following operations on Project table.  
2. Insert 2. Delete 3. Exit  
  
 \*/  
  
import java.sql.\*;  
import java.util.Scanner;  
  
public class Q27Project {  
  
  
 public static void main(String[] args) throws Exception{  
  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; // String jdbc\_url = "jdbc:postgresql://172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
 Class.forName("com.mysql.cj.jdbc.Driver"); // " Class.forName("org.postgresql.Driver";  
 Connection con = DriverManager.getConnection(jdbc\_url, user, pwd);  
 Scanner sc = new Scanner(System.in);  
  
 String query = "create table project(id int,name varchar(100),status varchar(100))";  
  
 Statement st = con.createStatement();  
 st.executeUpdate(query);  
  
  
 int choice;  
 do {  
 System.out.println("1. Insert \n 2.delete \n 3. Exit");  
 System.out.println("Enter ur choice:");  
 choice = sc.nextInt();  
  
 switch(choice)  
 {  
 case 1:  
 String query2 = "insert into project(id,name,status)values(?,?,?)";  
 PreparedStatement pst = con.prepareStatement(query2);  
  
 System.out.println("Enter the id:");  
 int id = sc.nextInt();  
 System.out.println("Enter the name:");  
 String name = sc.next();  
 System.out.println("Enter the status [complete/NotComplete]:");  
 String status = sc.next();  
  
 pst.setInt(1,id);  
 pst.setString(2,name);  
 pst.setString(3,status);  
 pst.executeUpdate();  
 break;  
  
 case 2 :  
 String search = "delete from project where id =?";  
 PreparedStatement pst1 = con.prepareStatement(search);  
 System.out.println("Enter the project id which u want to be delete:");  
 int pid = sc.nextInt();  
  
 pst1.setInt(1,pid);  
 int delete = pst1.executeUpdate();  
 System.out.println("no of rows deleted:"+delete);  
 break;  
  
  
 }  
 }while(choice!=3);  
  
 }  
}  
  
/\*  
1. Insert  
 2.delete  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
1  
Enter the name:  
swapnil  
Enter the status [complete/NotComplete]:  
NotComplete  
1. Insert  
 2.delete  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
2  
Enter the name:  
omu  
Enter the status [complete/NotComplete]:  
complete  
1. Insert  
 2.delete  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
3  
Enter the name:  
avi  
Enter the status [complete/NotComplete]:  
complete  
1. Insert  
 2.delete  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
44  
Enter the name:  
wednesday  
Enter the status [complete/NotComplete]:  
complete  
1. Insert  
 2.delete  
 3. Exit  
Enter ur choice:  
2  
Enter the project id which u want to be delete:  
2  
no of rows deleted:1  
1. Insert  
 2.delete  
 3. Exit  
  
 +------+-----------+-------------+  
| id | name | status |  
+------+-----------+-------------+  
| 1 | swapnil | NotComplete |  
| 3 | avi | complete |  
| 44 | wednesday | complete |  
+------+-----------+-------------+  
 \*/

/\*  
28) Write a java program for the following:  
i. To create a Product(Pid, Pname, Price) table.  
 ii. Insert at least five records into the table.  
 iii. Display all the records from a table.  
 \*/  
  
import java.sql.\*;  
import java.util.Scanner;  
  
public class Q28Product {  
 public static void main(String[] args) throws Exception {  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; // String jdbc\_url = "jdbc:postgresql://172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
 Class.forName("com.mysql.cj.jdbc.Driver"); // " Class.forName("org.postgresql.Driver";  
 Connection con = DriverManager.getConnection(jdbc\_url, user, pwd);  
 Scanner sc = new Scanner(System.in);  
  
 String createTable = "create table product(id int, name varchar(100),price int)";  
 Statement st = con.createStatement();  
 int created = st.executeUpdate(createTable);  
 if (created == 0)  
 System.out.println("Table Product is created...");  
  
 int choice;  
 do {  
 System.out.println("1. Insert \n 2.display \n 3. Exit");  
 System.out.println("Enter ur choice:");  
 choice = sc.nextInt();  
  
 switch (choice) {  
 case 1:  
 String query2 = "insert into product(id,name,price)values(?,?,?)";  
 PreparedStatement pst = con.prepareStatement(query2);  
  
 System.out.println("Enter the id:");  
 int id = sc.nextInt();  
 System.out.println("Enter the name:");  
 String name = sc.next();  
 System.out.println("Enter the price:");  
 int price = sc.nextInt();  
  
 pst.setInt(1, id);  
 pst.setString(2, name);  
 pst.setInt(3, price);  
 pst.executeUpdate();  
 break;  
  
 case 2:  
 String select = "select \* from product";  
 Statement st1 = con.createStatement();  
 ResultSet rs = st1.executeQuery(select);  
  
 while (rs.next()) {  
 System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getInt(3));  
 }  
 break;  
  
  
 }  
 } while (choice != 3);  
 }  
}  
  
/\* OUTPUT:  
Table Product is created...  
1. Insert  
 2.display  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
1  
Enter the name:  
mobile  
Enter the price:  
15000  
1. Insert  
 2.display  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
2  
Enter the name:  
laptop  
Enter the price:  
50000  
1. Insert  
 2.display  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
3  
Enter the name:  
shirt  
Enter the price:  
500  
1. Insert  
 2.display  
 3. Exit  
Enter ur choice:  
1  
Enter the id:  
5  
Enter the name:  
soap  
Enter the price:  
10  
1. Insert  
 2.display  
 3. Exit  
Enter ur choice:  
2  
1 mobile 15000  
2 laptop 50000  
3 shirt 500  
5 soap 10  
1. Insert  
 2.display  
 3. Exit  
Enter ur choice:  
  
 \*/

/\*  
29) Write a Java program to delete the details of given employee (ENo EName Salary).  
Accept employee ID through command line. (Use PreparedStatement Interface)  
 \*/  
  
import java.sql.\*;  
public class Q29Delete {  
 public static void main(String[] args) throws Exception{  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; //String jdbc\_url = "jdbc:postgresql//172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
  
 Class.forName("com.mysql.cj.jdbc.Driver"); // Class.forName("org.postgresql.Driver");  
  
 int id = Integer.parseInt(args[0]);  
 // int id = 2;  
 String query = "delete from employee1 where eid = '"+id+"'";  
 Connection con = DriverManager.getConnection(jdbc\_url,user,pwd);  
 Statement st = con.createStatement();  
 int delete = st.executeUpdate(query);  
 System.out.println("No of rows deleted:"+delete);  
  
 }  
}  
  
/\* OUTPUT:  
MariaDB [java\_jdbc]> select \* from employee1;  
+------+---------+---------+  
| eid | ename | esalary |  
+------+---------+---------+  
| 1 | swapnil | 8999 |  
| 2 | omu | 999 |  
| 3 | avi | 888 |  
+------+---------+---------+  
  
PS C:\Users\swapniljamble\IdeaProjects\MyProject\src> javac Q29Delete.java  
PS C:\Users\swapniljamble\IdeaProjects\MyProject\src> java Q29Delete 2  
No of rows deleted:1  
  
MariaDB [java\_jdbc]> select \* from employee1;  
+------+---------+---------+  
| eid | ename | esalary |  
+------+---------+---------+  
| 1 | swapnil | 8999 |  
| 3 | avi | 888 |  
+------+---------+---------+  
2 rows in set (0.001 sec)  
  
 \*/

/\*30) Write a Java Program for the implementation of scrollable ResultSet.  
 Assume Teacher table with attributes (TID, TName, Salary) is already created.  
 \*/  
  
import java.sql.\*;  
  
public class Q30Scroll {  
  
  
 public static void main(String[] args) throws Exception {  
 String jdbc\_url = "jdbc:mysql://localhost:3306/java\_jdbc"; //String jdbc\_url = "jdbc:postgresql//172.16.6.1/ty50";  
 String user = "root"; // String user = "ty50";  
 String pwd = "";  
  
 Class.forName("com.mysql.cj.jdbc.Driver"); // Class.forName("org.postgresql.Driver");  
 Connection con = DriverManager.getConnection(jdbc\_url,user,pwd);  
  
 Statement st = con.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);  
  
 ResultSet rs = st.executeQuery("select \* from teacher");  
 System.out.println("RECORDS IN THE TABLE...");  
 while (rs.next()) {  
 System.out.println(rs.getInt(1) + " -> " + rs.getString(2)+" "+rs.getInt(3));  
 }  
  
 rs.first();  
 System.out.println("FIRST RECORD...");  
 System.out.println(rs.getInt(1) + " " + rs.getString(2)+" "+rs.getInt(3));  
  
 rs.absolute(3);  
 System.out.println("THIRD RECORD...");  
 System.out.println(rs.getInt(1) + " -> " + rs.getString(2)+" "+rs.getInt(3));  
  
 rs.last();  
 System.out.println("LAST RECORD...");  
 System.out.println(rs.getInt(1) + " -> " + rs.getString(2)+" "+rs.getInt(3));  
  
 rs.previous();  
 rs.relative(-1);  
 System.out.println("LAST TO FIRST RECORD...");  
 System.out.println(rs.getInt(1) + " -> " + rs.getString(2)+" "+rs.getInt(3));  
 con.close();  
 }  
  
}