**Collections Assignment**

**Q-1. Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.**

**package** collections;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Scanner;

**public class** Q1 {

**public static void** main(String[] args) {

List<Float> ls=**new** ArrayList<>();

Scanner sc=**new** Scanner(System.***in***);

Float num;

**char** c;

**do**{

System.***out***.println(**"Enter element to add in your array list"**);

ls.add(sc.nextFloat());

System.***out***.println(**"Enter 1 to continue"**);

System.***out***.println(**"Enter 2 to add the float numbers"**);

c=sc.next().charAt(0);

} **while**(c==**'1'**);

Iterator<Float> itr=ls.iterator();

**float** sum=0.0f;

Float add=**new** Float(sum);

**while**(itr.hasNext()){

add+=itr.next();

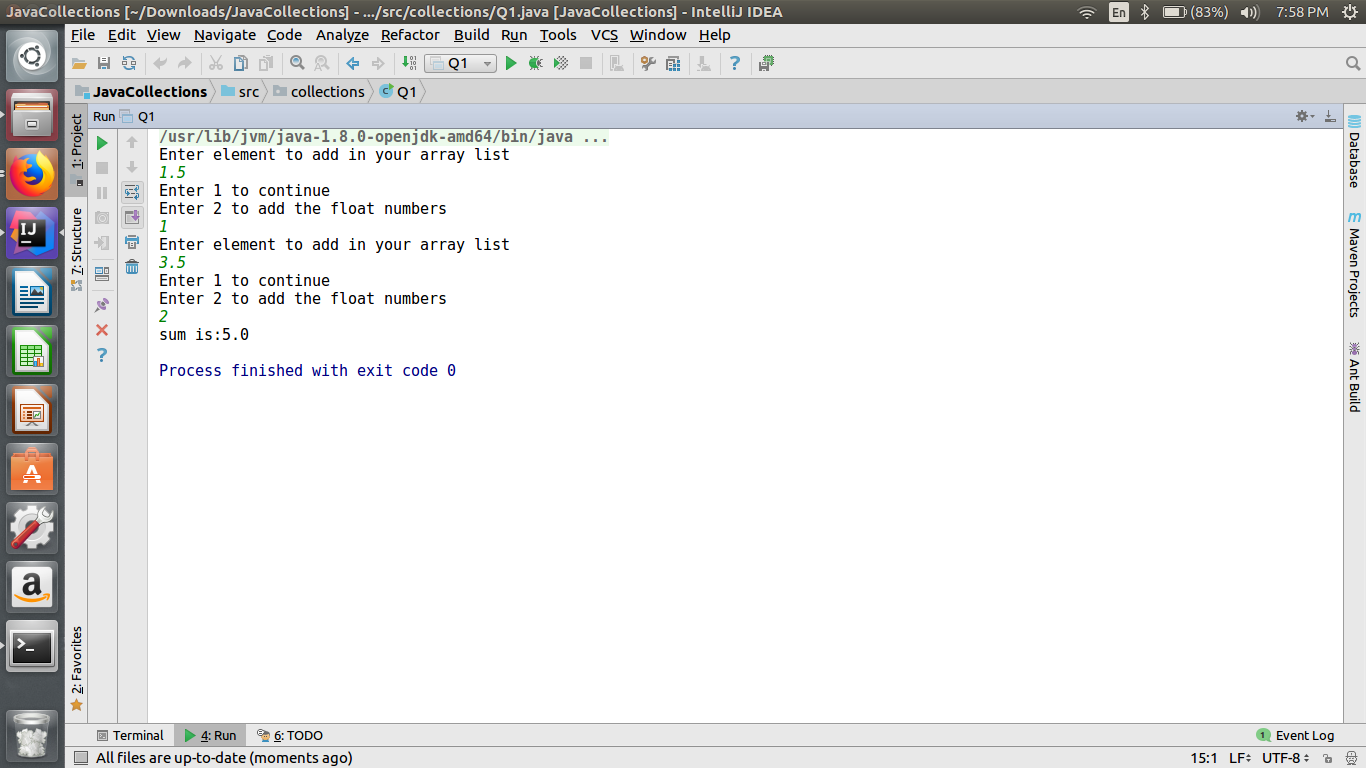
}

System.***out***.println(**"sum is:"**+add);

}

}

**OUTPUT:-**

****

**Q-2. Write a method that takes a string and returns the number of unique characters in the string.**

**package collections;**

**import java.util.HashMap;**

**import java.util.Iterator;**

**import java.util.Map;**

**import java.util.Scanner;**

**public class Q2 {**

**Scanner sc=new Scanner(System.*in*);**

**int uniqueCharacter(){**

**String str;**

**System.*out*.println("Enter a String");**

**str=sc.next();**

**Map<Character,Integer> hm=new HashMap<>();**

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

**if(hm.containsKey(str.charAt(i))){**

**hm.put(str.charAt(i), hm.get(str.charAt(i))+1 );**

**}**

**else{**

**hm.put(str.charAt(i),1);**

**}**

**}**

**int num=0;**

**Iterator<Character> itr=hm.keySet().iterator();**

**while(itr.hasNext()){**

**Character key=itr.next();**

**if(hm.get(key).equals(1)){**

**num++;**

**}**

**}**

**return num;**

**}**

**public static void main(String[] args) {**

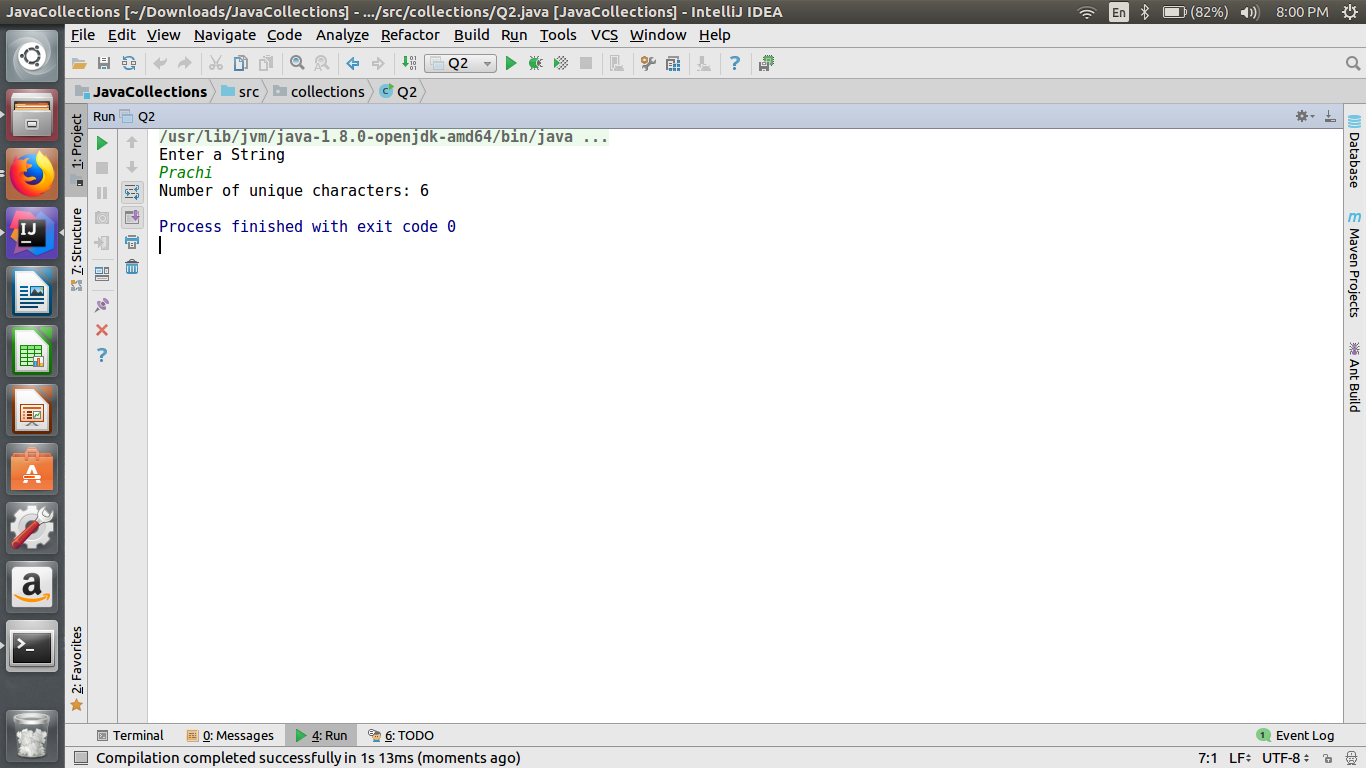
**Q2 obj=new Q2();**

**System.*out*.println("Number of unique characters: "+obj.uniqueCharacter());**

**}**

**}**

**OUTPUT:-**

****

**Q-3. Write a method that takes a string and print the number of occurrence of each character characters in the string.**

**package collections;**

**import java.util.HashMap;**

**import java.util.Iterator;**

**import java.util.Map;**

**import java.util.Scanner;**

**public class Q3 {**

**Scanner sc=new Scanner(System.*in*);**

**void occurencesOfcharacters(){**

**String str;**

**System.*out*.println("Enter a String");**

**str=sc.nextLine();**

**Map<Character,Integer> hm=new HashMap<>();**

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

**if(hm.containsKey(str.charAt(i))){**

**hm.put(str.charAt(i), hm.get(str.charAt(i))+1 );**

**}**

**else{**

**hm.put(str.charAt(i),1);**

**}**

**}**

**Iterator<Character> itr=hm.keySet().iterator();**

**while(itr.hasNext()){**

**Character key=itr.next();**

**System.*out*.println("Character "+key+" Occurences "+hm.get(key));**

**}**

**}**

**public static void main(String[] args) {**

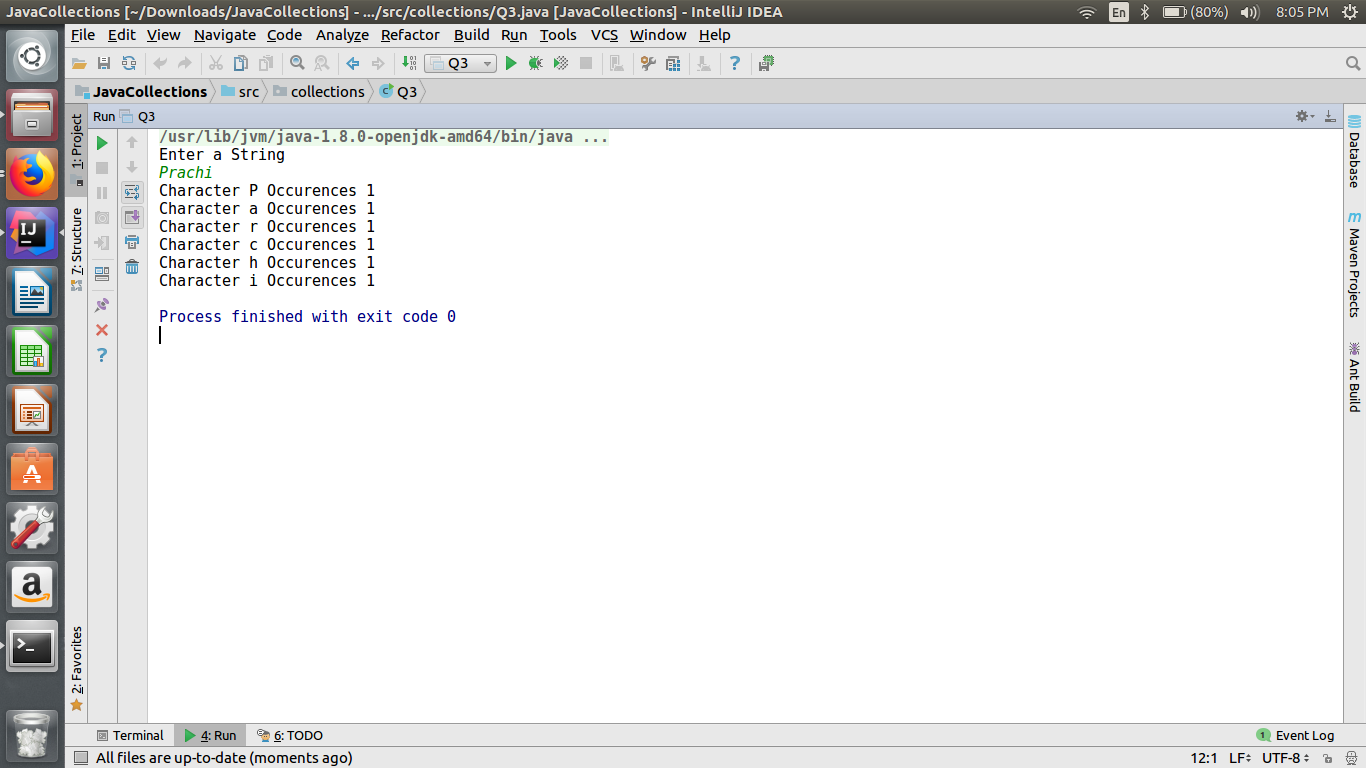
**Q3 obj=new Q3();**

**obj.occurencesOfcharacters();**

**}**

**}**

**OUTPUT:-**

****

**Q-4. Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name.**

**package collections;**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.Comparator;**

**import java.util.List;**

**public class Q4 {**

**public static void main(String[] args) {**

**List al=new ArrayList();**

**al.add(new Employee("prachi",.5,23.5));**

**al.add(new Employee("megha",97.5,22.5));**

**al.add(new Employee("yati",7.5,22.5));**

**al.add(new Employee("yashika",1.5,23.5));**

**Collections.*sort*(al,new Salary());**

**for(int i=0;i<al.size();i++)**

**{**

**Employee e=(Employee)al.get(i);**

**System.*out*.println(e.name+" "+e.salary+" "+e.age);**

**}**

**}**

**}**

**class Employee {**

**String name;**

**double salary=0.0,age=0.0;**

**public Employee(String name,double salary,double age) {**

**this.name=name;**

**this.salary=salary;**

**this.age=age;**

**}**

**}**

**class Salary implements Comparator<Employee> {**

**@Override**

**public int compare(Employee e1, Employee e2) {**

**if (e1.salary > e2.salary) {**

**return -1;**

**} else {**

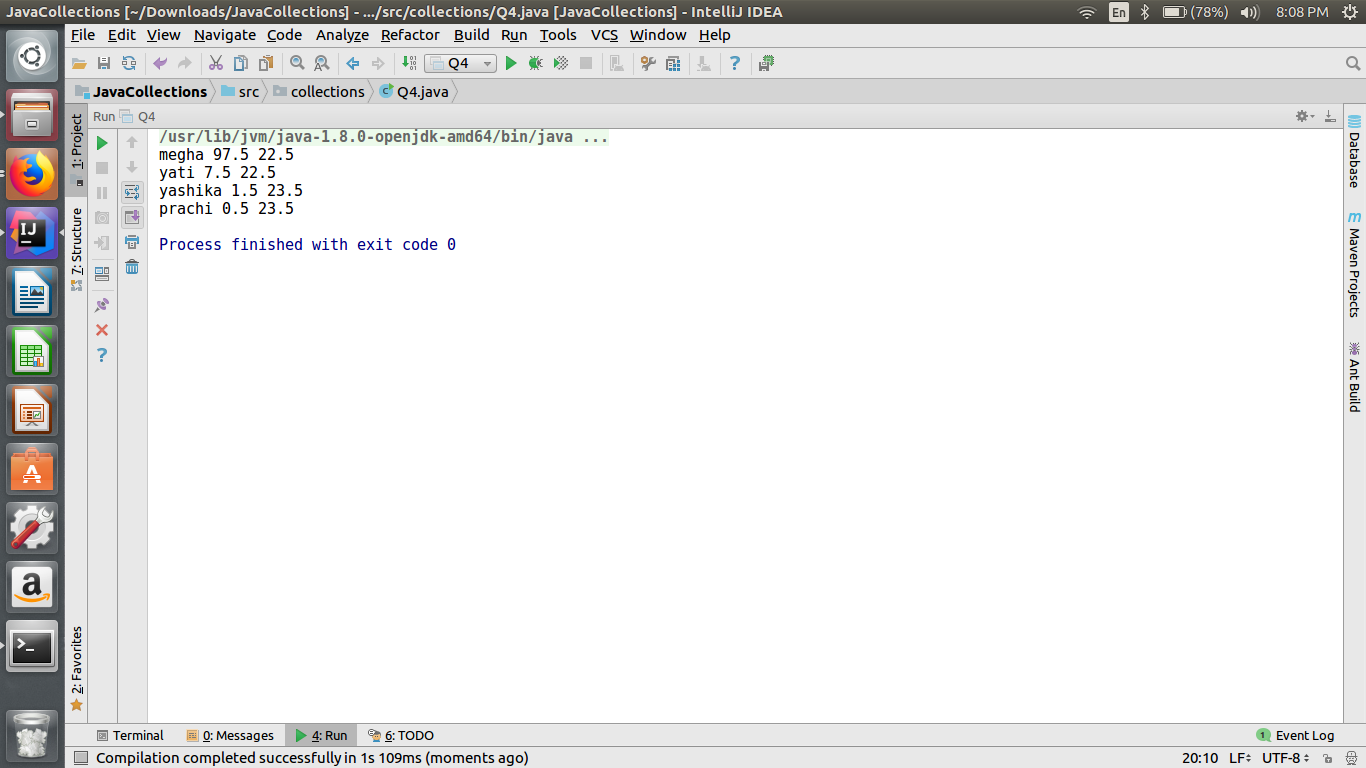
**return 1;**

**}**

**}**

**}**

**OUTPUT:-**

****

**Q-5. Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age.**

**package collections;**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.Comparator;**

**class Student {**

**String name="";**

**double score=0.0,age=0.0;**

***// static ArrayList al;***

**public Student(String name,double score,double age) {**

**this.name=name;**

**this.score=score;**

**this.age=age;**

**}**

**}**

**public class Q5{**

**public static void main(String[] args) {**

**ArrayList al=new ArrayList();**

**al.add(new Student("prachi",97.5,23.5));**

**al.add(new Student("megha",97.5,22.5));**

**al.add(new Student("yati",7.5,22.5));**

**al.add(new Student("yashika",7.5,23.5));**

**Collections.*sort*(al,new score());**

**for(int i=0;i<al.size();i++)**

**{**

**Student s=(Student)al.get(i);**

**System.*out*.println(s.name+" "+s.score+" "+s.age);**

**}**

**}**

**}**

**class score implements Comparator<Student> {**

**@Override**

**public int compare(Student o1, Student o2) {**

**if (o1.score == o2.score) {**

**return o1.name.compareTo(o2.name);**

**} else if (o1.score > o2.score) {**

**return 1;**

**} else {**

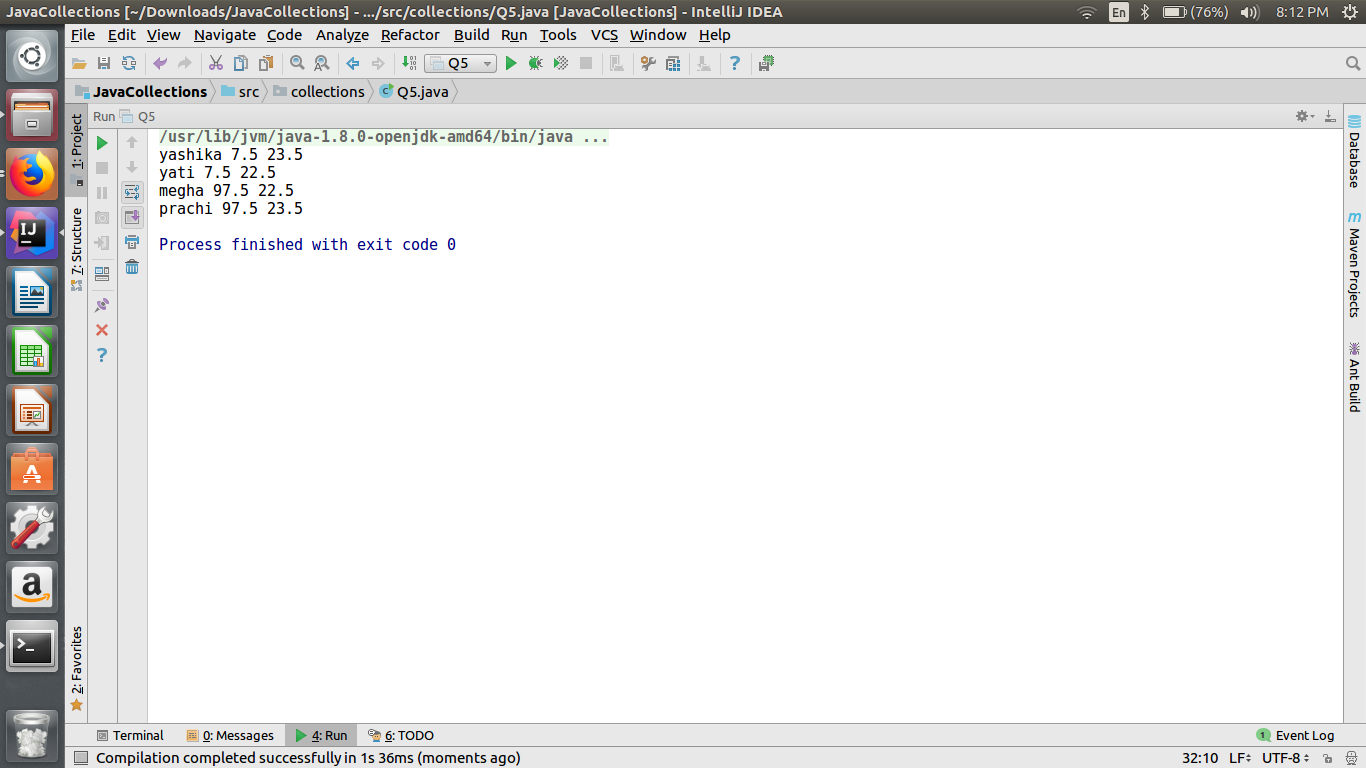
**return -1;**

**}**

**}**

**}**

**OUTPUT:-**

****

**Q-6. Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.**

**import java.util.\*;**

**public class Q6 {**

**public static void main(String[] args) {**

**int n[]={9,1,9,1,4,5,6,8,8,1,1,2,2,11,4,11,6};**

**HashMap<Integer,Integer> fre=new HashMap();**

**HashMap<Integer,Integer> in=new HashMap();**

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

**{**

**if(fre.containsKey(n[i]))**

**{**

**fre.put(n[i], fre.get(n[i])+1);**

**}**

**else**

**{**

**fre.put(n[i],1);**

**in.put(n[i], i);**

**}**

**}**

**List<Map.Entry<Integer,Integer>> key=new ArrayList<>(fre.entrySet());**

**Collections.*sort*(key , new comp(in));**

**for(Map.Entry<Integer,Integer> i : key)**

**{**

**System.*out*.println(i);**

**}**

**}**

**}**

**class comp implements Comparator <Map.Entry<Integer,Integer>>**

**{**

**HashMap<Integer,Integer> m;**

**public comp(HashMap<Integer,Integer> obj)**

**{**

**m=obj;**

**}**

**@Override**

**public int compare(Map.Entry<Integer,Integer> o1 , Map.Entry<Integer,Integer> o2) {**

**if(o1.getValue().equals(o2.getValue()))**

**{**

**return m.get(o1.getKey())-m.get(o2.getKey());**

**}**

**else**

**{**

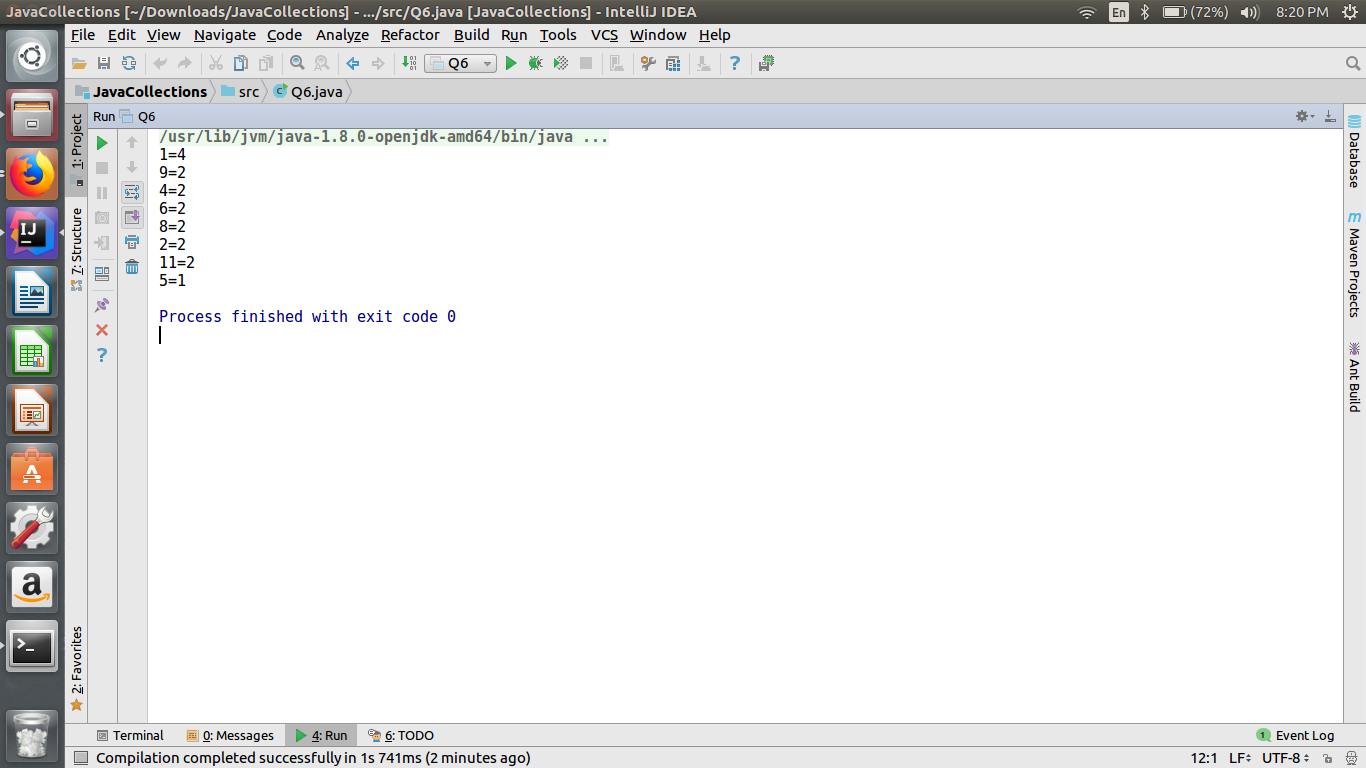
**return o2.getValue().compareTo(o1.getValue());**

**}**

**}**

**}**

**OUTPUT:-**

****

**Q-7. Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity ­ O(1)).**

**import java.util.Scanner;**

**import java.util.Stack;**

**public class Q7 {**

**public static void main(String[] args) {**

**Scanner sc=new Scanner(System.*in*);**

**int size=4;**

**SpecialStack obj=new SpecialStack(size);**

**int c=0;**

**do {**

**System.*out*.println("Enter 1 to push");**

**System.*out*.println("Enter 2 to pop");**

**System.*out*.println("Enter 3 to find the stack isempty");**

**System.*out*.println("Enter 4 to find the stack is full");**

**System.*out*.println("Enter 5 to find minimum in the stack");**

**System.*out*.println("Enter 6 to exit");**

**System.*out*.println("Enter your choice");**

**c=sc.nextInt();**

**switch(c){**

**case 1:System.*out*.println("Enter a number to push");**

**obj.push(sc.nextInt());**

**break;**

**case 2:**

**System.*out*.println("Popped Element is:"+obj.pop());**

**break;**

**case 3:**

**obj.isEmpty();**

**break;**

**case 4:**

**obj.isFull();**

**break;**

**case 5:**

**System.*out*.println("Minimum Element on stack is:"+ obj.minStack.peek());**

**break;**

**case 6:**

**System.*out*.println("exit");**

**break;**

**default:**

**System.*out*.println("Exit");**

**}**

**} while(c!=6);**

**}**

**}**

**class SpecialStack{**

**int top;**

**int size;**

**int[] stack;**

**Stack<Integer> minStack = new Stack<>();**

**int min=0;**

**public SpecialStack(int arraySize){**

**size=arraySize;**

**stack= new int[size];**

**top=-1;**

**}**

**public void isFull(){**

**if(top==size-1)**

**System.*out*.println("full");**

**else**

**System.*out*.println("not full");**

**}**

**public void isEmpty(){**

**if(top==-1)**

**System.*out*.println("Stack is empty");**

**else**

**System.*out*.println("Stack is not empty");**

**}**

**public void push(int value){**

**if(top==size-1){**

**System.*out*.println("Stack is full, can't push a value");**

**}**

**else if(top==-1 && minStack.isEmpty()){**

**top=top+1;**

**stack[top]=value;**

**minStack.push(value);**

**}**

**else if(value < minStack.peek()){**

**top=top+1;**

**stack[top]=value;**

**minStack.push(value);**

**}**

**else**

**{**

**top=top+1;**

**stack[top]=value;**

**minStack.push(minStack.peek());**

**}**

**}**

**public int pop(){**

**int num=stack[top];**

**top=top-1;**

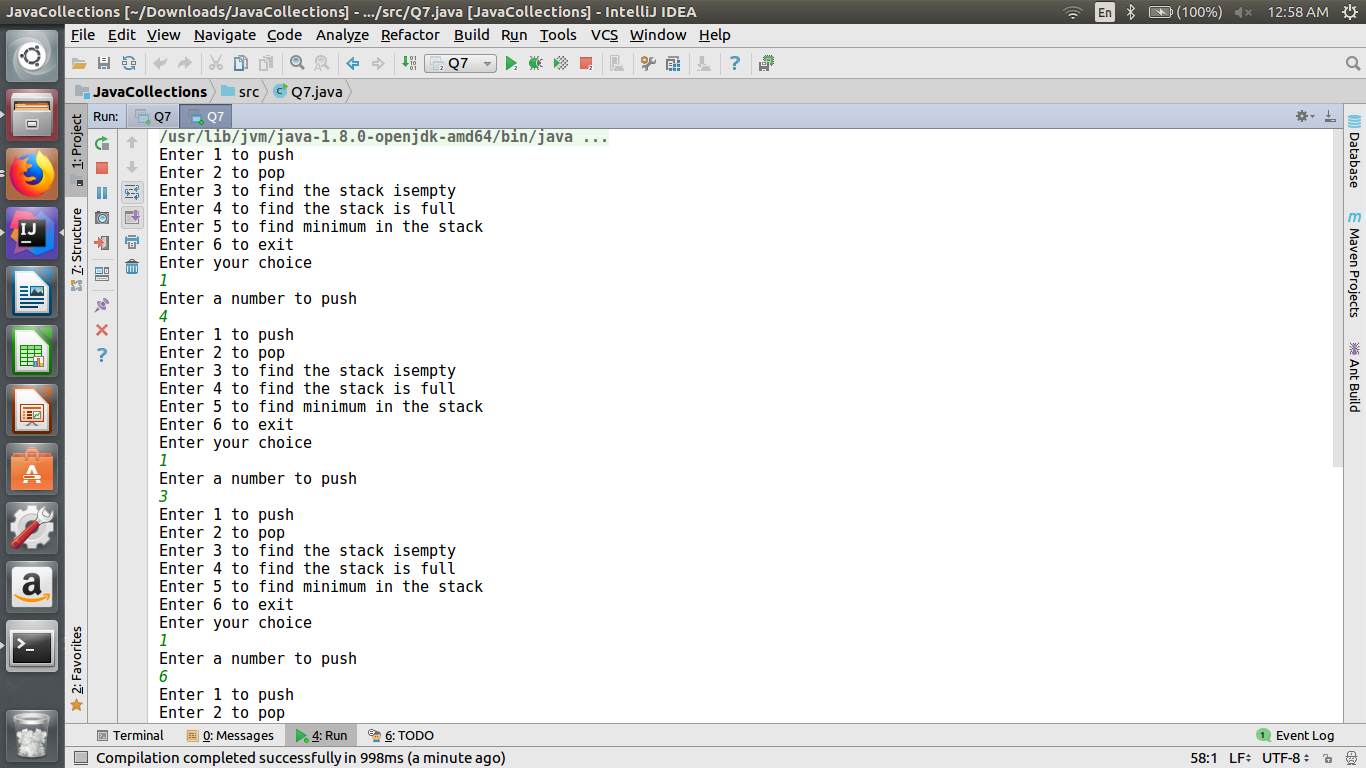
**minStack.pop();**

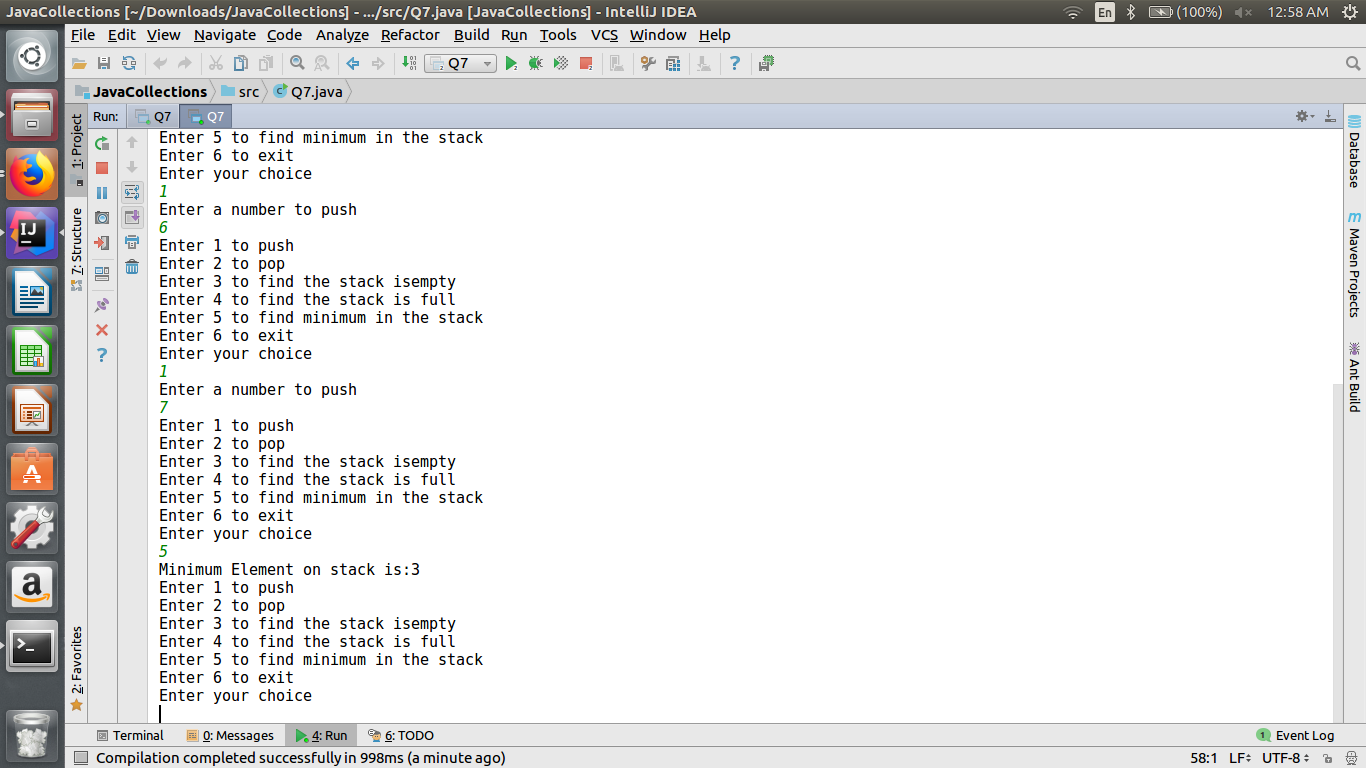
**return num;**

**}**

**}**

**OUTPUT:-**

****

****

**Q-8. Write a program to format date as example "21-March-2016".**

**package collections;**

**import java.text.SimpleDateFormat;**

**import java.util.Date;**

**public class Q8 {**

**public static void main(String[] args) {**

**Date today = new Date();**

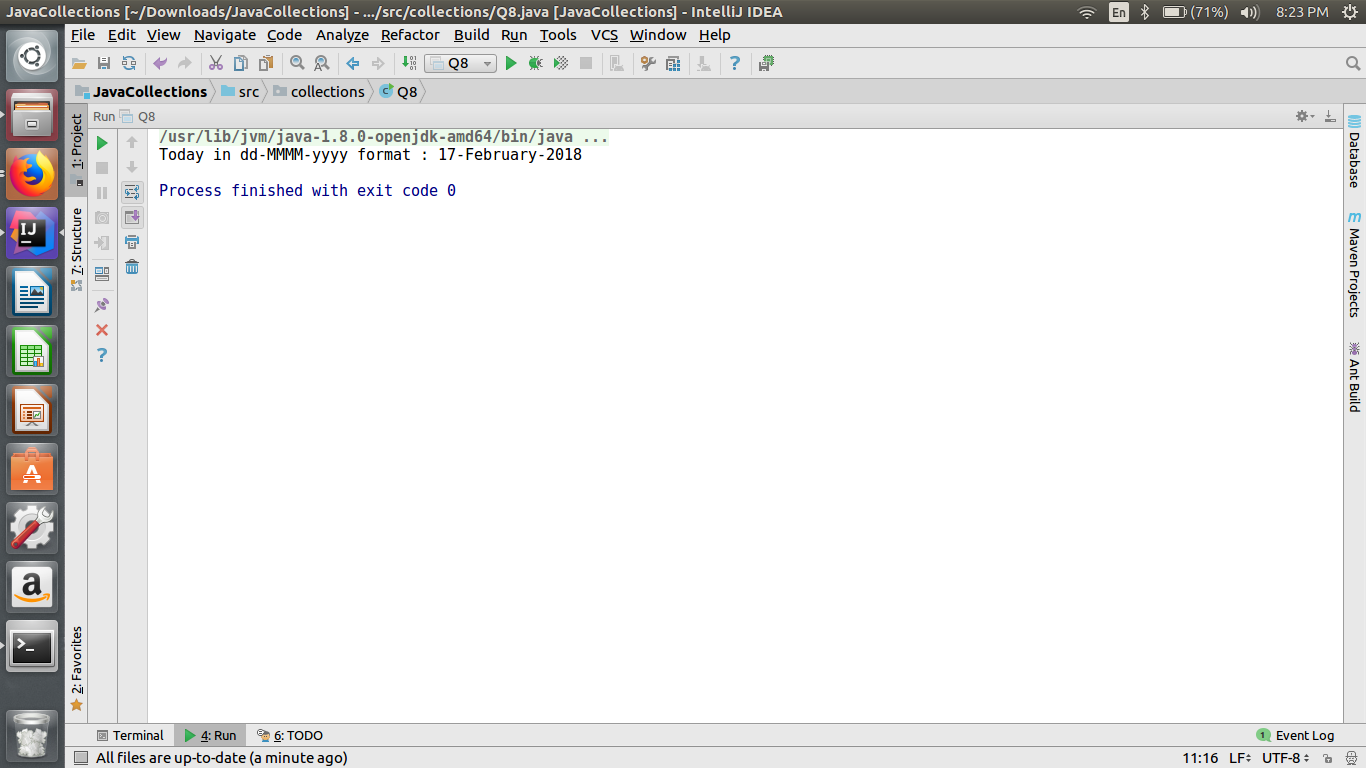
**SimpleDateFormat DATE\_FORMAT = new SimpleDateFormat("dd-MMMM-yyyy");**

**System.*out*.println("Today in dd-MMMM-yyyy format : " + DATE\_FORMAT.format(today));**

**}**

**}**

**OUTPUT:-**

****

**Q-9. Write a program to display times in different country format.**

**package collections;**

**import java.text.DateFormat;**

**import java.util.Date;**

**import java.util.Locale;**

**public class Q9 {**

**public static void main(String[] args) {**

**Date d1 = new Date();**

**DateFormat german = DateFormat.*getDateInstance* (DateFormat.*FULL*, Locale.*GERMANY*);**

**DateFormat china = DateFormat.*getDateInstance* (DateFormat.*FULL*, Locale.*CHINA*);**

**DateFormat france = DateFormat.*getDateInstance* (DateFormat.*FULL*, Locale.*FRANCE*);**

**DateFormat taiwan = DateFormat.*getDateInstance* (DateFormat.*FULL*, Locale.*TAIWAN*);**

**DateFormat us = DateFormat.*getDateInstance* (DateFormat.*FULL*, Locale.*US*);**

**System.*out*.println("today is in Italian Language in German Format : "+ german.format(d1));**

**System.*out*.println("today is in Italian Language in china Format : "+ china.format(d1) );**

**System.*out*.println("today is in Italian Language in Switzerland Format : "+ france.format(d1));**

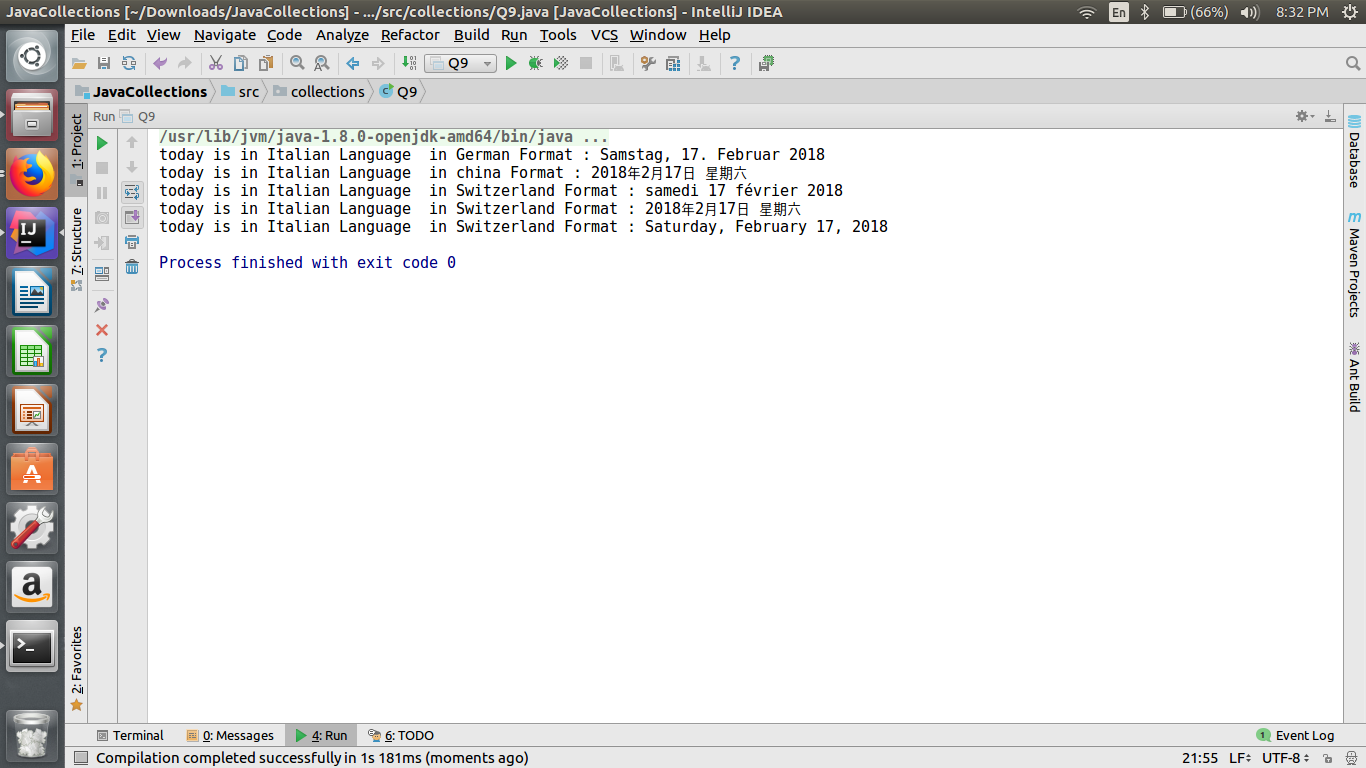
**System.*out*.println("today is in Italian Language in Switzerland Format : "+ taiwan.format(d1));**

**System.*out*.println("today is in Italian Language in Switzerland Format : "+ us.format(d1));**

**}**

**}**

**OUTPUT:-**

****