**Collections And Date**

**Ans 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.

\*/

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

**import** java.util.ArrayList;

**public class** Q1List {

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

ArrayList<Float> list = **new** ArrayList<Float>();

list.add(4.669F);

list.add(8.5F);

list.add(5.7F);

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

{

**float** sum = 0;

**for** (**float** i : list) {

sum += i;

}

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

}

}

}

**Output:** 18.869

**Ans 2.**

**/\***Write a method that takes a string and returns the number of unique characters in the string.

**\*/**

**import** java.util.HashMap;

**import** java.util.Scanner;

**public class** Q2UniqueCharacters {

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

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

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

String string = scanner.next();

**int** stringLength=string.length();

**char**[] charArray = string.toCharArray();

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

**for** (**int** i = 0; i < stringLength; i++) {

**char** ch = charArray[i];

**if** (!hashmap.containsKey(ch)) {

hashmap.put(ch,1);

}

**else**

{

**int** getValue = hashmap.get(ch);

getValue++;

hashmap.put(ch,getValue);

}

}

**for** (Character key : hashmap.keySet()) {

**if**(hashmap.get(key)==1)

{

System.***out***.println(**"Iterating map"**);

System.***out***.println(**"unique key is "**+key);

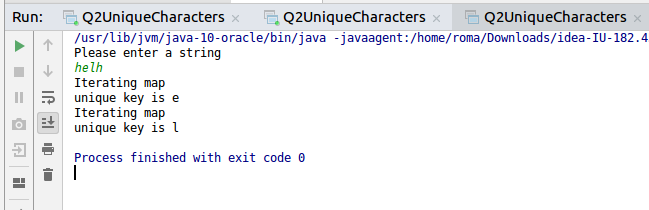
}

}

}

}

Output



**Ans 3.**

**/\***Write a method that takes a string and print the number of occurrence of each character characters in the string

**\*/**

import java.util.HashMap;

import java.util.Scanner;

public class Q3CountOccurances {

public static void main(String[] args) {

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

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

String string = scanner.next();

int stringLength=string.length();

char[] charArray = string.toCharArray();

HashMap<Character,Integer> hashmap = new HashMap<>();

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

char ch = charArray[i];

if (!hashmap.containsKey(ch)) {

hashmap.put(ch,1);

}

else

{

int getValue = hashmap.get(ch);

getValue++;

hashmap.put(ch,getValue);

}

}

for (Character key : hashmap.keySet()) {

System.*out*.println("Iterating map");

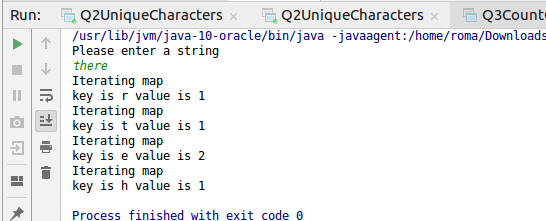
System.*out*.println("key is "+key + " value is "+hashmap.get(key));

}

}

}

Output

****

**Ans 4.**

/\*Write a program to sort Employee objects based on highest salary using Comparator.

\* Employee class{ Double Age; Double Salary; String Name

/

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

**class** Q4Employee {

**private** String **name**;

**private int id**;

**private** String **competancy**;

**private** Double **salary**;

**public** Q4Employee(String name, **int** id, String competancy, Double salary) {

**this**.**name** = name;

**this**.**id** = id;

**this**.**competancy** = competancy;

**this**.**salary** = salary;

}

@Override

**public** String toString() {

**return "\n Q4Employee{"** +

**"name='"** + **name** + **'\''** +

**", id="** + **id** +

**", competancy='"** + **competancy** + **'\''** +

**", salary="** + **salary** +

**'}'**;

}

**public** Double getSalary() {

**return salary**;

}

**public void** setSalary(Double salary) {

**this**.**salary** = salary;

}

**public** String getName() {

**return name**;

}

**public void** setName(String name) **throws** Exception {

**if** (name == **null**) {

**this**.**name** = name;

}

}

**public int** getId() {

**return id**;

}

**public void** setId(**int** id) **throws** Exception {

**if** (id > 0) {

**this**.**id** = id;

}

}

**public** String getCompetancy() {

**return competancy**;

}

**public void** setCompetancy(String competancy) {

**this**.**competancy** = competancy;

}

}

**public class** Q4Main

{

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

List<Q4Employee> list= **new** ArrayList<Q4Employee>();

list.add(**new** Q4Employee(**"Roma"**,3035,**"JVM"**,35000.0));

list.add( **new** Q4Employee(**"Richa"**,30358,**"JVM"**,15000.0));

Comparator<Q4Employee> comparator=**new** Comparator<Q4Employee> () {

@Override

**public int** compare(Q4Employee employee, Q4Employee employee1) {

**if**(employee.getSalary()>employee1.getSalary())

{

**return** -1;

}

**else if**(employee.getSalary()<employee1.getSalary())

{

**return** 1;

}

**else**

**return** 0;

}

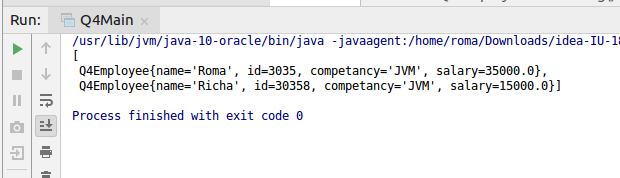
};

Collections.*sort*(list,comparator);

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

}

}



**Ans 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

\*/

**import** java.util.ArrayList;

**import** java.util.Comparator;

**import** java.util.List;

**class** Q5Student

{

**private** String **name**;

**private double score**;

**private double age**;

**public** Q5Student(String name, **double** score, **double** age) {

**this**.**name** = name;

**this**.**score** = score;

**this**.**age** = age;

}

@Override

**public** String toString() {

**return "\n Q5Student{"** +

**"name='"** + **name** + **'\''** +

**", score="** + **score** +

**", age="** + **age** +

**'}'**;

}

**public** String getName() {

**return name**;

}

**public void** setName(String name) {

**this**.**name** = name;

}

**public double** getScore() {

**return score**;

}

**public void** setScore(**double** score) {

**this**.**score** = score;

}

**public double** getAge() {

**return age**;

}

**public void** setAge(**double** age) {

**this**.**age** = age;

}

}

**public class** Q5Main

{

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

{

List<Q5Student> q5StudentList=**new** ArrayList<Q5Student>();

q5StudentList.add(**new** Q5Student(**"Roma"**,90.88,23.0));

q5StudentList.add(**new** Q5Student(**"Roma"**,70.88,29.0));

q5StudentList.add(**new** Q5Student(**"Rohan"**,90.88,21.0));

q5StudentList.add(**new** Q5Student(**"Richa"**,20.88,28.0));

System.***out***.println(q5StudentList);

Comparator<Q5Student> comparator= Comparator.*comparing*(Q5Student::getScore).thenComparing(Q5Student::getName);

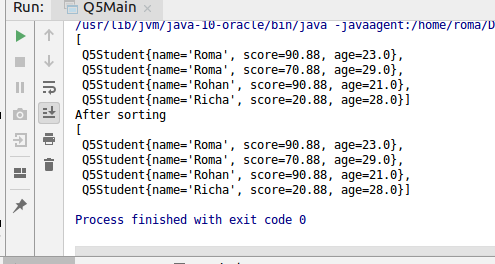
System.***out***.println(**"After sorting"**);

System.***out***.println(q5StudentList);

}

}

}



**Ans 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**[] arr = {3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6};

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

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

**if** (!(map.containsKey(arr[i]))) {

map.put(arr[i], 0);

}

**if** (map.containsKey(arr[i])) {

map.put(arr[i], map.get(arr[i]) + 1);

}

}

System.***out***.println(**"Before Sorting : "**);

**for** (Map.Entry<Integer, Integer> entry : map.entrySet()) {

System.***out***.println(**"Key : "** + entry.getKey() + **" Occurances : "** + entry.getValue());

}

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

sortedList.sort(**new** Comparator<Map.Entry<Integer, Integer>>() {

@Override

**public int** compare(Map.Entry<Integer, Integer> t1, Map.Entry<Integer, Integer> t2) {

**if** (t1.getValue() < t2.getValue())

**return** 1;

**else if** (t1.getValue() > t2.getValue())

**return** -1;

**else**

**return** t2.getKey().compareTo(t1.getKey());

}

});

System.***out***.println(**"After sorting : "**);

**for** (Map.Entry<Integer,Integer> integerEntry: sortedList){

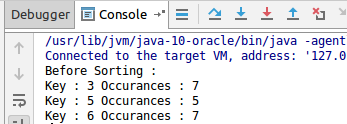
System.***out***.println(**"Key : "** + integerEntry.getKey() + **" Occurances : "** + integerEntry.getValue());

}

}

}

**Output**

****

**Ans 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.Stack;

**public class** Question7 **extends** Stack<Integer>

{

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

**void** push(**int** x){

**if**(isEmpty() == **true**){

**super**.push(x);

**integerStack**.push(x);

}**else**{

**super**.push(x);

**int** y = **integerStack**.pop();

**integerStack**.push(y);

**if**(x < y){

**integerStack**.push(x);

}

**else** {

**integerStack**.push(y);

}

}

}

**public** Integer pop(){

**int** x = **super**.pop();

**integerStack**.pop();

**return** x;

}

**int** getMin(){

**int** x = **integerStack**.pop();

**integerStack**.pop();

**return** x;

}

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

Question7 ques = **new** Question7();

ques.push(10);

ques.push(20);

ques.push(30);

System.***out***.println(**"Elements in stack "**+ques);

System.***out***.println(**"Minimum "**+ques.getMin());

ques.push(62);

ques.push(2);

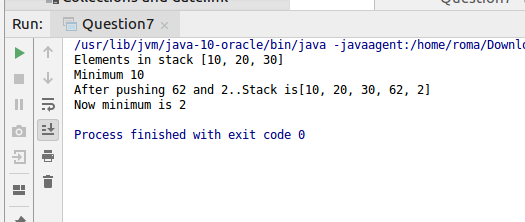
System.***out***.println(**"After pushing 62 and 2..Stack is"**+ques);

System.***out***.println(**"Now minimum is "**+ques.getMin());

}

}

Output



**Ans 8.**

/\*Write a program to format date as example "21-March-2016"

\*/h

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public class** Q8DateFormat {

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

Date date = **new** Date();

SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"dd-MMMM-yyyy"**);

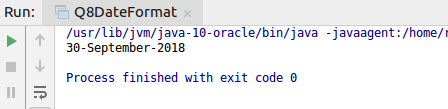
String dateToStr = simpleDateFormat.format(date);

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

}

}

Output



**Ans 9.**

/\*Write a program to display times in different country format.

\*/

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.time.LocalTime;

import java.time.ZoneId;

import java.util.Date;

import java.util.Locale;

import java.util.TimeZone;

public class Q9TimeConutries {

public static void main(String[] args) {

SimpleDateFormat indiasdf = new SimpleDateFormat("dd/MM/yyyy zZ HH:mm:ss a");

SimpleDateFormat usasdf = new SimpleDateFormat("MM/dd/yyyy zZ HH:mm:ss a");

SimpleDateFormat paksdf = new SimpleDateFormat("yyyy-MM-dd zZ HH:mm:ss a");

TimeZone ind = TimeZone.*getTimeZone*("Asia/Kolkata");

TimeZone usa = TimeZone.*getTimeZone*("US/Central");

TimeZone pakistan = TimeZone.*getTimeZone*("Asia/Karachi");

indiasdf.setTimeZone(ind);

usasdf.setTimeZone(usa);

paksdf.setTimeZone(pakistan);

System.*out*.println("India Time format "+indiasdf.format(new Date()));

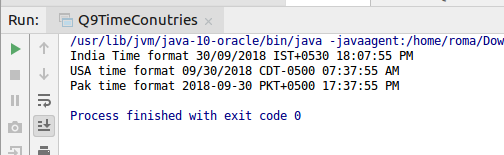
System.*out*.println("USA time format "+usasdf.format(new Date()));

System.*out*.println("Pak time format "+paksdf.format(new Date()));

}

}

**Output**

****