

Rajalakshmi Engineering College

Name: Amruthavalli M

Email: 240701037@rajalakshmi.edu.in

Roll no: 240701037

Phone: 9384914691

Branch: REC

Department: CSE - Section 8

Batch: 2028

Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 9_MCQ

Attempt : 1

Total Mark : 15

Marks Obtained : 14

Section 1 : MCQ

1. What is the correct way to create an ArrayList in Java?

Answer

`ArrayList<String> list = new ArrayList<>();`

Status : Correct

Marks : 1/1

2. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
```

```
list.add(3);  
list.add(4);  
list.set(2, 10);  
System.out.println(list);  
}  
}
```

Answer

[1, 2, 10, 4]

Status : Correct

Marks : 1/1

3. What does the addFirst() method of LinkedList do?

Answer

Adds an element to the beginning of the list

Status : Correct

Marks : 1/1

4. What will be the output of the following code?

```
import java.util.*;  
class Main {  
    public static void main(String[] args) {  
        ArrayList<Integer> list = new ArrayList<>();  
        list.add(10);  
        list.add(20);  
        list.add(30);  
        list.remove(1);  
        System.out.println(list);  
    }  
}
```

Answer

[10, 30]

Status : Correct

Marks : 1/1

5. What will be the output of the following code?

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        list.remove("Apple");
        System.out.println(list);
    }
}
```

Answer

[Banana]

Status : Correct

Marks : 1/1

6. What will be the output of the following code?

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("apple");
        list.add("banana");
        list.add("cherry");
        list.add("banana");
        System.out.println(list.lastIndexOf("banana"));
    }
}
```

Answer

3

Status : Correct

Marks : 1/1

7. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Java");
        list.add("Python");
        list.add("Java");
        list.add("C++");
        System.out.println(list.indexOf("Java"));
    }
}
```

Answer

0

Status : Correct

Marks : 1/1

8. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.add(4);
        list.add(5);
        System.out.println(list.get(3));
    }
}
```

Answer

4

Status : Correct

Marks : 1/1

9. What will be the output of the following code?

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Stack<Integer> s = new Stack<>();
        s.push(10);
        s.push(20);
        s.push(30);
        System.out.println(s.peek());
    }
}
```

Answer

30

Status : Correct

Marks : 1/1

10. What will be the output of the following code?

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Stack<Integer> stack = new Stack<>();
        for (int i = 1; i <= 3; i++)
            stack.push(i * 2);
        stack.pop();
        stack.push(10);
        System.out.println(stack.peek());
    }
}
```

Answer

10

Status : Correct

Marks : 1/1

11. What will be the output of the following code?

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(10);
        list.add(20);
        list.add(30);
        System.out.println("Size of the list: " + list.size());
    }
}
```

Answer

Size of the list: 3

Status : Correct

Marks : 1/1

12. How can you access the first element of an ArrayList named as list?

Answer

list.get(0);

Status : Correct

Marks : 1/1

13. Which method is used to add an element to the top of the stack?

Answer

push()

Status : Correct

Marks : 1/1

14. Which of the following methods removes and returns the last element from a LinkedList?

Answer

pop()

Status : Wrong

Marks : 0/1

15. What is Collection in Java?

Answer

A group of objects

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: Amruthavalli M

Email: 240701037@rajalakshmi.edu.in

Roll no: 240701037

Phone: 9384914691

Branch: REC

Department: CSE - Section 8

Batch: 2028

Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q1

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bobby is tasked with processing a sequence of numbers from a monitoring system. He needs to extract a strictly increasing subsequence using an ArrayList. The program should dynamically add numbers to the ArrayList only if they are greater than the last number currently stored in the list. Bobby aims to efficiently utilize the dynamic resizing and indexing features of the ArrayList to solve this problem.

Help Bobby implement this solution.

Input Format

The first line of input consists of an integer N, representing the number of elements.

The second line consists of N space-separated integers, representing the elements.

Output Format

The output prints the list of integers in increasing sequence, ignoring out-of-order elements.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7

3 5 9 1 11 7 13

Output: [3, 5, 9, 11, 13]

Answer

```
import java.util.*;
class Main{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int N=s.nextInt();
        ArrayList<Integer>list=new ArrayList<>();
        for(int i=0;i<N;i++){
            int num=s.nextInt();
            if(list.isEmpty()||(num>list.get(list.size()-1))){
                list.add(num);
            }
        }
        System.out.println(list);
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Amruthavalli M

Email: 240701037@rajalakshmi.edu.in

Roll no: 240701037

Phone: 9384914691

Branch: REC

Department: CSE - Section 8

Batch: 2028

Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q2

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Vikram loves listening to music and wants to create a simple playlist manager using Java Collections. The playlist supports the following operations:

"ADD <song>" Adds the song to the end of the playlist. "REMOVE <song>" Removes the first occurrence of the song from the playlist. If the song is not found, do nothing. "SHOW" Displays all songs in the playlist in order. If the playlist is empty, print "EMPTY". "NEXT" Moves to the next song in the playlist and prints its name. If the playlist is empty, print "EMPTY".

The playlist maintains a "current song" position that starts at the first song when it's added. The NEXT command moves to the next song and prints it, wrapping around to the first song after reaching the last song. When removing songs, the current position adjusts accordingly to maintain

proper navigation.

Help Vikram implement this playlist manager.

Input Format

The first line of the input consists of an integer n, the number of operations.

The next n lines, each containing a command:

- "ADD <song>"
- "REMOVE <song>"
- "SHOW"
- "NEXT"

Output Format

For each "SHOW" command, print the songs in order, separated by spaces.

For each "NEXT" command, print the next song in the playlist.

If no song exists, print "EMPTY".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7

ADD song1

ADD song2

SHOW

NEXT

REMOVE song2

SHOW

NEXT

Output: song1 song2

song2

song1

song1

Answer

```
import java.util.*;
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        sc.nextLine();
```

```
        LinkedList<String> playlist = new LinkedList<>();  
        int currentIndex = -1;
```

```
        for (int i = 0; i < n; i++) {  
            String command = sc.nextLine().trim();
```

```
            if (command.startsWith("ADD")) {  
                String song = command.split(" ")[1];  
                playlist.add(song);
```

```
            }  
            if (currentIndex == -1) {  
                currentIndex = 0;  
            }  
        }
```

```
        else if (command.startsWith("REMOVE")) {  
            String song = command.split(" ")[1];  
            int indexToRemove = playlist.indexOf(song);
```

```
            if (indexToRemove != -1) {  
                playlist.remove(indexToRemove);
```

```
                // adjust currentIndex after removal  
                if (playlist.isEmpty()) {  
                    currentIndex = -1;  
                } else if (indexToRemove < currentIndex) {  
                    currentIndex--;  
                } else if (indexToRemove == currentIndex) {  
                    if (currentIndex >= playlist.size()) {  
                        currentIndex = 0; // wrap if needed  
                    }  
                }  
            }  
        }
```

```
    }
```

```
}  
  
    else if (command.equals("SHOW")) {  
        if (playlist.isEmpty()) {  
            System.out.print("EMPTY ");  
        } else {  
            for (String song : playlist) {  
                System.out.print(song + " ");  
            }  
        }  
    }  
  
    else if (command.equals("NEXT")) {  
        if (playlist.isEmpty()) {  
            System.out.print("EMPTY ");  
        } else {  
            currentIndex = (currentIndex + 1) % playlist.size();  
            System.out.print(playlist.get(currentIndex) + " ");  
        }  
    }  
  
    sc.close();  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Amruthavalli M
Email: 240701037@rajalakshmi.edu.in
Roll no: 240701037
Phone: 9384914691
Branch: REC
Department: CSE - Section 8
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Assist Pranitha in developing a program that takes an integer N as input, representing the number of names to be read. Then read N names and store them in an ArrayList. Finally, input a search string and output the frequency of that string in the list of names.

Note: Some parts of the code are provided as snippets, and you need to complete the remaining sections by writing the necessary code.

Input Format

The first line of input consists of an integer N, representing the number of names to be read.

The following N lines consist of N names, as a string.

The last line consists of a string, representing the name to be searched.

Output Format

The output prints a single integer, representing the frequency of the specified name in the given list.

If the specified name is not found, print 0.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

Alice

Bob

Ankit

Alice

Pranitha

Alice

Output: 2

Answer

```
import java.util.*;
class Main{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int N=s.nextInt();
        s.nextLine();
        ArrayList<String>names=new ArrayList<>();
        for(int i=0;i<N;i++){
            names.add(s.nextLine());
        }
        String search=s.nextLine();
        int c=0;
        for(String name:names){
            if(name.equals(search)){
                c++;
            }
        }
        System.out.println(c);
    }
}
```

}
}
Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Amruthavalli M

Email: 240701037@rajalakshmi.edu.in

Roll no: 240701037

Phone: 9384914691

Branch: REC

Department: CSE - Section 8

Batch: 2028

Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 10_MCQ

Attempt : 1

Total Mark : 15

Marks Obtained : 15

Section 1 : MCQ

1. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("A", 1);
        map.put("B", 2);
        map.put("C", 3);
        System.out.println(map.containsKey("B"));
    }
}
```

Answer

true

Status : Correct

Marks : 1/1

2. What will happen if you add a null element to a TreeSet?

Answer

An exception occurs

Status : Correct

Marks : 1/1

3. Which method retrieves the lowest key in a TreeMap?

Answer

firstKey()

Status : Correct

Marks : 1/1

4. How does HashSet check for duplicate elements?

Answer

Using equals() and hashCode()

Status : Correct

Marks : 1/1

5. Which of the following is true about TreeMap?

Answer

It maintains natural ordering

Status : Correct

Marks : 1/1

6. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
```

```
map.put("X", 10);
map.put("Y", 20);
map.put("Z", 30);
map.remove("Y");
System.out.println(map);
}
}
```

Answer

{X=10, Z=30}

Status : Correct

Marks : 1/1

7. What will happen if you add elements in descending order in a TreeSet?

Answer

They are sorted in ascending order

Status : Correct

Marks : 1/1

8. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        HashMap<String, String> map = new HashMap<>();
        map.put("A", "Apple");
        map.put("B", "Banana");
        map.put("C", "Cherry");
        map.replace("B", "Blueberry");
        System.out.println(map);
    }
}
```

Answer

{A=Apple, B=Blueberry, C=Cherry}

Status : Correct

Marks : 1/1

9. Which method removes all elements from a Set?

Answer

clear()

Status : Correct

Marks : 1/1

10. Which of the following is true about HashMap?

Answer

It is not synchronized

Status : Correct

Marks : 1/1

11. What is the time complexity of retrieving an element from a HashSet?

Answer

$O(1)$

Status : Correct

Marks : 1/1

12. What happens if two keys have the same hash code in a HashMap?

Answer

A linked list is used to store values with the same hash

Status : Correct

Marks : 1/1

13. What happens when you add duplicate elements to a HashSet?

Answer

The duplicate is ignored

Status : Correct

Marks : 1/1

14. Which statement is true about HashSet and TreeSet?

Answer

TreeSet provides sorted elements

Status : Correct

Marks : 1/1

15. Which of the following allows null keys in Java?

Answer

HashMap

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: Amruthavalli M

Email: 240701037@rajalakshmi.edu.in

Roll no: 240701037

Phone: 9384914691

Branch: REC

Department: CSE - Section 8

Batch: 2028

Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q1

Attempt : 1

Total Mark : 10

Marks Obtained : 0

Section 1 : COD

1. Problem Statement

A city traffic management system needs to track vehicles entering a toll booth. Each vehicle is uniquely identified by its registration number. The system should allow adding vehicles to a record, ensuring that no duplicate registration numbers exist. The vehicles should be stored in a HashSet, which does not guarantee any specific order.

Your task is to implement a program using a HashSet that allows adding vehicle details and displaying the records.

Input Format

The first line of input contains an integer N - the number of vehicles.

The next N lines contain details of each vehicle in the format: "RegNumber

OwnerName VehicleType"

1. RegNumber (String) - A unique registration number (Alphanumeric).
2. OwnerName (String) - The name of the vehicle owner.
3. VehicleType (String, Car, Bike, or Truck) - The type of vehicle.

If a vehicle with the same registration number is already present, ignore the duplicate entry.

Output Format

The output prints the unique vehicle records in any order (since HashSet does not maintain order).

Output format: "RegNumber OwnerName VehicleType"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

KA01AB1234 John Car
MH02CD5678 Alice Bike
DL03EF9012 Bob Truck
TN04GH3456 Mike Car
KA01AB1234 John Car

Output: TN04GH3456 Mike Car
KA01AB1234 John Car
MH02CD5678 Alice Bike
DL03EF9012 Bob Truck

Answer

```
import java.util.*;
class Vehicle{
    String reg,owner,type;
    Vehicle(String r,String o,String t){
        reg=r;
        owner=o;
        type=t;
    }
    public boolean equals(Object o)
```

```

    {
        return (o instanceof Vehicle V && reg.equals(V.reg));
    }
    public int hashCode(){
        return reg.hashCode();
    }
    public String toString()
    {
        return reg + " " +owner + " "+type;
    }
    class Main{
        public static void main(String[] args){
            Scanner s=new Scanner(System.in);
            int n=s.nextInt();
            s.nextLine();
            Set<Vehicle>set=new HashSet<>();
            for(int i=0;i<n;i++){
                String[]a=s.nextLine().split(" ");
                set.add(new Vehicle(a[0],a[1],a[2]));
            }
            set.forEach(System.out::println);
        }
    }
}

```

Status : Wrong

Marks : 0/10

Rajalakshmi Engineering College

Name: Amruthavalli M

Email: 240701037@rajalakshmi.edu.in

Roll no: 240701037

Phone: 9384914691

Branch: REC

Department: CSE - Section 8

Batch: 2028

Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q2

Attempt : 1

Total Mark : 10

Marks Obtained : 6.5

Section 1 : COD

1. Problem Statement

John is organizing a fruit festival, and the quantities of various fruits are stored in a HashMap where fruit names are keys and quantities are values.

Help him develop a program to find the total quantity of fruits for the festival by summing up the values in the HashMap.

Input Format

The input consists of fruit quantities in the format 'fruitName:quantity', where fruitName is the name of the fruit(a string), and quantity is a double value representing the quantity.

The input is terminated by entering "done".

Output Format

The output prints a double value, representing the sum of values in the HashMap, rounded off to two decimal places.

If the value is not numeric, print "Invalid input".

If any special characters other than ':' are entered, print "Invalid format".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Banana:15.2

Orange:56.3

Mango:47.3

done

Output: 118.80

Answer

```
import java.util.*;
class Main{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        HashMap<String,Double>map=new HashMap<>();
        while(true){
            String input=s.nextLine().trim();
            if(input.equalsIgnoreCase("done")){
                break;
            }
            String[] parts=input.split(":");
            String name=parts[0];
            double qty=Double.parseDouble(parts[1]);
            map.put(name,qty);
        }
        double total=0.0;
        for(double v:map.values()){
            total+=v;
        }
        System.out.printf("%.2f",total);
    }
}
```

Status : Partially correct

Marks : 6.5/10

Rajalakshmi Engineering College

Name: Amruthavalli M
Email: 240701037@rajalakshmi.edu.in
Roll no: 240701037
Phone: 9384914691
Branch: REC
Department: CSE - Section 8
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

Priya is analyzing encrypted messages in a research project. She wants to analyze the frequency of each character in a given paragraph. The characters should be stored in a TreeMap so that the output is sorted in ascending order of characters automatically.

You are required to build a Java program that:

Uses a `TreeMap<Character, Integer>` to count how many times each character appears in the message. Ignores spaces and considers only alphabets (case-sensitive). Outputs the frequencies of characters in sorted order.

You must use a TreeMap in the class named MessageAnalyzer.

Input Format

The first line of input contains an integer n, the number of lines in the message.

The next n lines each contain a string (the encrypted message line).

Output Format

The first line of output prints: "Character Frequency:"

Then print each character and its frequency in the format: "<character>: <count>"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
Hello World
Java

Output: Character Frequency:

H: 1

J: 1

W: 1

a: 2

d: 1

e: 1

l: 3

o: 2

r: 1

v: 1

Answer

```
import java.util.*;
class Main{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int n=Integer.parseInt(s.nextLine());
        TreeMap<Character,Integer>map=new TreeMap<>();
        for(int i=0;i<n;i++){
            String line=s.nextLine();
            for(char ch:line.toCharArray()){
                if(Character.isLetter(ch)){
                    map.put(ch,map.getDefault(ch,0)+1);
                }
            }
        }
    }
}
```

```
    }  
    }  
    }  
    System.out.println("Character Frequency:");  
    for(Character c:map.keySet()){  
        System.out.println(c+": "+map.get(c));  
    }  
    }  
}
```

Status : Correct

Marks : 10/10