

# Rajalakshmi Engineering College

Name: Varnika V

Email: 240701578@rajalakshmi.edu.in

Roll no: 240701578

Phone: 9384321919

Branch: REC

Department: CSE - Section 10

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**

- 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

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 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

4. 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

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

Answer

Adds an element to the beginning of the list

Status : Correct

Marks : 1/1

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

Answer

push()

Status : Correct

Marks : 1/1

7. 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

8. 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

9. What is Collection in Java?

**Answer**

A group of interfaces

Status : Wrong

Marks : 0/1

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

Answer

list.get(0);

Status : Correct

Marks : 1/1

11. 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

12. 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**

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

**Answer**

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

**Status : Correct**

**Marks : 1/1**

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

**Answer**

removeLast()

**Status : Correct**

**Marks : 1/1**

15. 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

# Rajalakshmi Engineering College

Name: Varnika V

Email: 240701578@rajalakshmi.edu.in

Roll no: 240701578

Phone: 9384321919

Branch: REC

Department: CSE - Section 10

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.ArrayList;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        ArrayList<Integer> increasingList = new ArrayList<>();

        for (int i = 0; i < n; i++) {
            int number = sc.nextInt();
            if (increasingList.isEmpty() || number >
increasingList.get(increasingList.size() - 1)) {
                increasingList.add(number);
            }
        }

        System.out.println(increasingList);
        sc.close();
    }
}
```

**Status : Correct**

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Varnika V

Email: 240701578@rajalakshmi.edu.in

Roll no: 240701578

Phone: 9384321919

Branch: REC

Department: CSE - Section 10

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.LinkedList;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = Integer.parseInt(sc.nextLine());
        LinkedList<String> playlist = new LinkedList<>();
        int currentIndex = 0;
        for (int i = 0; i < n; i++) {
            String commandLine = sc.nextLine();
            String[] parts = commandLine.split(" ", 2);
            String command = parts[0];
            if (command.equals("ADD")) {
                playlist.add(parts[1]);
            } else if (command.equals("REMOVE")) {
                String songToRemove = parts[1];
                int index = playlist.indexOf(songToRemove);
                if (index != -1) {
                    playlist.remove(songToRemove);
                }
                if (index < currentIndex || currentIndex >= playlist.size()) {
                    currentIndex = Math.max(0, currentIndex - 1);
                }
            }
        }
        } else if (command.equals("SHOW")) {
            if (playlist.isEmpty()) {
                System.out.println("EMPTY");
            } else {
                for (String song : playlist) {
                    System.out.print(song + " ");
                }
                System.out.println();
            }
        } else if (command.equals("NEXT")) {
            if (playlist.isEmpty()) {
                System.out.println("EMPTY");
            } else {
                currentIndex = (currentIndex + 1) % playlist.size();
                System.out.println(playlist.get(currentIndex));
            }
        }
    }
}
```

}

**Status : Correct**

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Varnika V

Email: 240701578@rajalakshmi.edu.in

Roll no: 240701578

Phone: 9384321919

Branch: REC

Department: CSE - Section 10

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.ArrayList;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = Integer.parseInt(sc.nextLine());
        ArrayList<String> names = new ArrayList<>();

        for (int i = 0; i < N; i++) {
            names.add(sc.nextLine());
        }

        String searchName = sc.nextLine();
        int count = 0;

        for (String name : names) {
            if (name.equals(searchName)) {
```

```
        count++;  
    }  
    System.out.println(count);  
}  
}
```

**Status : Correct**

**Marks : 10/10**