Rajalakshmi Engineering College

Name: Jovitha Sheethal B
Email: 241001099@rajalakshmi.edu.in
Roll no: 241001099
Phone: 6385334710
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

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 : 12

Section 1 : MCQ

1. What is Collection in Java?

Answer

A group of objects

Status : Correct *Marks : 1/1*

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

Answer

push()

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<String> list = new ArrayList<>();
        list.add("apple");
        list.add("banana");
        list.add("cherry");
        list.add("banana");
        System.out.println(list.lastIndexOf("banana"));
    }
}
```

Answer

4

Status : Wrong ***Marks : 0/1***

4. 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***

5. 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***

6. 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***

7. 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

[20, 30]

Status : Wrong ***Marks : 0/1***

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

Answer

list.get(0);

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> 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***

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

Answer

ArrayList&lt;String&gt; list = new ArrayList&lt;&gt;();

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> s = new Stack<>();
      s.push(10);
      s.push(20);
      s.push(30);
      System.out.println(s.peek());
   }
}
```

Answer

30

Status : Correct ***Marks : 1/1***

12. 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***

13. 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***

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

Answer

Adds an element to the beginning of the list

Status : Correct ***Marks : 1/1***

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

Answer

removeFirst()

Status : Wrong ***Marks : 0/1***

Rajalakshmi Engineering College

Name: Jovitha Sheethal B
Email: 241001099@rajalakshmi.edu.in
Roll no: 241001099
Phone: 6385334710
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

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.*;

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 num = sc.nextInt();


         if (increasingList.isEmpty()) {
            increasingList.add(num);
         }

         else if (num > increasingList.get(increasingList.size() - 1)) {
            increasingList.add(num);
         }
      }

      System.out.print(increasingList);
```

```
        sc.close();
    }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Jovitha Sheethal B
Email: 241001099@rajalakshmi.edu.in
Roll no: 241001099
Phone: 6385334710
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

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.*;

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

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

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

      for (int i = 0; i < names.size(); i++) {
         if (names.get(i).equals(searchName)) {
                count++;
            }
        }

        System.out.print(count);
        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Jovitha Sheethal B
Email: 241001099@rajalakshmi.edu.in
Roll no: 241001099
Phone: 6385334710
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 9_PAH

Attempt : 1
Total Mark : 30
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Rekha is a teacher who wants to calculate the average of marks scored by her students in a test. She needs to store all the marks dynamically because the number of students may vary each time. Using an ArrayList allows her to easily add any number of marks without worrying about the initial size.

Help her implement the task.

Input Format

The first line of input is an integer n, representing the number of students..

The second line of input consists of n double values, representing the marks of each student, separated by a space.

Output Format

The output prints: "Average of the list: " followed by the average value formatted to two decimal places.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
1.0 2.0 3.0 4.0 5.0

Output: Average of the list: 3.00

Answer

```
// You are using Java
import java.util.*;

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

        ArrayList<Double> marksList = new ArrayList<>();
        double sum = 0.0;

        for (int i = 0; i < n; i++) {
            double mark = sc.nextDouble();
            marksList.add(mark);
            sum += mark;
        }

        double average = sum / n;
        System.out.printf("Average of the list: %.2f", average);

        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Aditi is analyzing stock market trends and wants to find the Next Greater Element (NGE) for each stock price in a list. The Next Greater Element for an element x in an array is the first element to the right that is greater than x. If no greater element exists, return -1 for that position.

Your task is to help Aditi by efficiently computing the Next Greater Element for each element in the given array using a Stack.

Example:

Input:

6

4 5 2 10 8 6

Output:

5 10 10 -1 -1 -1

Explanation:

For each element:

4   5 (next greater element)5   102   1010   -1 (No greater element)8   -16   -1

Input Format

The first line contains an integer n, representing the number of elements.

The second line contains n space-separated integers arr[i], where arr[i] is the stock price on the i-th day.

Output Format

The output prints n space-separated integers representing the Next Greater Element for each element in the array.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 6
4 5 2 10 8 6
Output: 5 10 10 -1 -1 -1

Answer

```
// You are using Java
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int[] arr = new int[n];
      for (int i = 0; i < n; i++) {
         arr[i] = sc.nextInt();
      }

      int[] nge = new int[n];
      Stack<Integer> stack = new Stack<>();

      // Traverse from right to left
      for (int i = n - 1; i >= 0; i--) {
         // Pop all smaller elements
         while (!stack.isEmpty() && stack.peek() <= arr[i]) {
            stack.pop();
         }

         // If stack is empty, no greater element exists
         if (stack.isEmpty()) {
            nge[i] = -1;
         } else {
            nge[i] = stack.peek();
         }

         // Push current element for future comparisons
         stack.push(arr[i]);
      }

      // Print the result
      for (int i = 0; i < n; i++) {
         System.out.print(nge[i] + " ");
      }
```

```
        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Arun is building a task manager to keep track of tasks using a LinkedList. The task manager supports the following operations:

"ADD <task>"   Adds the given task to the end of the list."REMOVE"   Removes the first task from the list."SHOW"   Displays all tasks in the list in order. If the list is empty, print "EMPTY".

Help Arun implement this functionality using a LinkedList.

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 <task>"
- "REMOVE"
- "SHOW"

Output Format

For each "SHOW" command, the output prints the tasks in order, separated by spaces.

If no tasks exist, print "EMPTY".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

```
ADD homework
ADD project
SHOW
REMOVE
SHOW
```

Output: homework project
project

Answer

```
// You are using Java
import java.util.*;

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

        LinkedList<String> tasks = new LinkedList<>();

        for (int i = 0; i < n; i++) {
            String commandLine = sc.nextLine().trim();
            String[] parts = commandLine.split(" ", 2);
            String command = parts[0];

            if (command.equals("ADD")) {
                String task = parts[1];
                tasks.add(task);
            }
            else if (command.equals("REMOVE")) {
                if (!tasks.isEmpty()) {
                    tasks.removeFirst();
                }
            }
            else if (command.equals("SHOW")) {
                if (tasks.isEmpty()) {
                    System.out.print("EMPTY ");
                } else {
                    for (String t : tasks) {
                        System.out.print(t + " ");
                    }
                }
            }
        }

        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Jovitha Sheethal B
Email: 241001099@rajalakshmi.edu.in
Roll no: 241001099
Phone: 6385334710
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 9_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 28.5

Section 1 : Coding

1. Problem Statement

Raman, a computer science teacher, is responsible for registering students for his programming class. To streamline the registration process, he wants to develop a program that stores students' names and allows him to retrieve a student's name based on their index in the list.

Raman has decided to use an ArrayList to store the names of students, as it provides efficient dynamic resizing and indexing.

Write a program that enables Raman to input the names of students and fetch a student's name using the specified index. If the entered index is invalid, the program should return an appropriate message.

Input Format

The first line of input consists of an integer n, representing the number of students to register.

The next n lines of input consist of the names of each student, one by one.

The last line of input is an integer, representing the index (0-indexed) of the element to retrieve.

Output Format

If the index is valid (within the bounds of the ArrayList), print "Element at index [index]: " followed by the element (student name as string).

If the index is invalid, print "Invalid index".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
Alice
Bob
Ankit
Alice
Prajit
2
Output: Element at index 2: Ankit

Answer

```
// You are using Java
import java.util.*;

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

        for (int i = 0; i < n; i++) {
            students.add(sc.next());
        }
        int index = sc.nextInt();

        if (index >= 0 && index < students.size()) {
            System.out.print("Element at index " + index + ": " + students.get(index));
        } else {
            System.out.print("Invalid index");
        }

        sc.close();
    }
}
```

Status : Partially correct ***Marks : 8.5/10***

2. Problem Statement

A teacher is filtering a list of words provided by students. Some words contain too many vowels, making them difficult for a spelling competition. The teacher decides to remove all words that contain more than two vowels.

Help the teacher to implement it using ArrayList.

Input Format

The first line contains an integer N, representing the number of words in the list.

The next N lines contain a string representing the words (one per line).

Output Format

The output consists of words that contain two or less than two vowels, printed in the same order they appeared in the input. Each word is printed on a new line.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1

sri
Output: sri

Answer

```
import java.util.ArrayList;
import java.util.Scanner;



class VowelFilter {
   public static void filterWords(int n, Scanner sc) {
      ArrayList<String> validWords = new ArrayList<>();
      for (int i = 0; i < n; i++) {
         String word = sc.nextLine();
         if (countVowels(word) <= 2) {
            validWords.add(word);
         }
      }
      for (String word : validWords) {
         System.out.println(word);
      }
   }

   private static int countVowels(String word) {
      int count = 0;
      for (char ch : word.toCharArray()) {
         if (isVowel(ch)) {
            count++;
         }
      }
      return count;
   }

   private static boolean isVowel(char ch) {
      return "aeiou".indexOf(ch) != -1;
   }
}

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

```
        VowelFilter.filterWords(n, sc);
        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Rahul is working on a list manipulation problem where he needs to reverse a specific subarray using a stack. Given an array and two indices l and r, he wants to reverse only the portion of the array from index l to r (both inclusive) while keeping the rest of the array unchanged.

Since Rahul wants to solve this problem efficiently, he decides to use a stack to reverse the subarray in O(r - l) time.

Your task is to help Rahul by implementing this functionality.

Input Format

The first line contains an integer n, the size of the array.

The second line contains n space-separated integers arr[i].

The third line contains two integers l and r, denoting the start and end indices of the subarray to reverse.

Note: The array follows 0-based indexing.

Output Format

The output prints the modified array after reversing the subarray between indices l and r.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 6
1 2 3 4 5 6
1 4
Output: 1 5 4 3 2 6

Answer

-

Status : Skipped ***Marks : 0/10***

4. Problem Statement

Rahul, a stock trader, wants to analyze the stock prices of a company over several days. For each day, he wants to determine the stock span, which is the number of consecutive days (including the current day) where the stock price is less than or equal to the price on that day.

The stock span helps him understand how long a stock has been continuously increasing or staying the same. You need to help Rahul by computing the stock span for each day using a Stack data structure efficiently.

Example:

Input:

7

100 80 60 70 60 75 85

Output:

1 1 1 2 1 4 6

Explanation:

For each day:

Day 1: Price = 100   Span = 1 (Only this day)Day 2: Price = 80   Span = 1 (Only this day)Day 3: Price = 60   Span = 1 (Only this day)Day 4: Price = 70   Span = 2 (Includes today and previous day)Day 5: Price = 60   Span = 1 (Only this day)Day 6: Price = 75   Span = 4 (Includes today and previous three days)Day 7: Price = 85   Span = 6 (Includes today and previous five days)

Input Format

The first line contains an integer n, the number of days.

The second line contains n space-separated integers prices[i], where prices[i] represents the stock price on the i-th day.

Output Format

The output prints n space-separated integers representing the stock span for each day.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7
100 80 60 70 60 75 85

Output: 1 1 1 2 1 4 6

Answer

```
// You are using Java
import java.util.*;

public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    int[] prices = new int[n];
    for (int i = 0; i < n; i++) {
      prices[i] = sc.nextInt();
    }

    int[] span = new int[n];
    Stack<Integer> stack = new Stack<>();

    for (int i = 0; i < n; i++) {
      while (!stack.isEmpty() && prices[stack.peek()] <= prices[i]) {
        stack.pop();
      }
```

```
            if (stack.isEmpty()) {
                span[i] = i + 1;
            } else {
                span[i] = i - stack.peek();
            }

            stack.push(i);
        }

        for (int i = 0; i < n; i++) {
            System.out.print(span[i] + " ");
        }

        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***