Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. PROBLEM STATEMENT:

Dave got two students who wants help with their doubt. Each handouts an integer and wants to find if one Integer Positive While the Other is Not Divisible by 3. Write a program to achieve this and conclude for them.

Input Format

The first line of input represents the first integer.

The second line of input represents the second integer.

Output Format

The output should display as "One of the integers is positive while the other is not divisible by 3." or "Neither of the integers meets the condition."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4
3
Output: One of the integers is positive while the other is not divisible by 3.

Answer

```
import java.util.Scanner;

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

      // Read the two integers from input
      int num1 = scanner.nextInt();
      int num2 = scanner.nextInt();

      boolean condition1 = (num1 > 0 && num2 % 3 != 0);
      boolean condition2 = (num2 > 0 && num1 % 3 != 0);

      if (condition1 || condition2) {
         System.out.println("One of the integers is positive while the other is not
divisible by 3.");
      } else {
         System.out.println("Neither of the integers meets the condition.");
      }

      scanner.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. PROBLEM STATEMENT:

Dave got two students who wants help with their doubt. Each handouts an integer and wants to find if one Integer Positive While the Other is Not Divisible by 3. Write a program to achieve this and conclude for them.

Input Format

The first line of input represents the first integer.

The second line of input represents the second integer.

Output Format

The output should display as "One of the integers is positive while the other is not divisible by 3." or "Neither of the integers meets the condition."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4
3
Output: One of the integers is positive while the other is not divisible by 3.

Answer

```
import java.util.Scanner;

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

      // Read the two integers from input
      int num1 = scanner.nextInt();
      int num2 = scanner.nextInt();

      boolean condition1 = (num1 > 0 && num2 % 3 != 0);
      boolean condition2 = (num2 > 0 && num1 % 3 != 0);

      if (condition1 || condition2) {
         System.out.println("One of the integers is positive while the other is not
divisible by 3.");
      } else {
         System.out.println("Neither of the integers meets the condition.");
      }

      scanner.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem statement

Manoj, a developer at MoneyMatters Inc., is working on improving the company's financial system. He needs to create a program that takes an integer input, converts it into a double, and displays both the original integer and the converted double value.

Input Format

The input consists of a single integer representing a monetary amount.

Output Format

The first line of the output displays the "Original Integer: ", followed by an integer representation of the input value.

The second line displays the "Converted Double: ", followed by a double value representing the input as a decimal value.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 20

Output: Original Integer: 20
Converted Double: 20.0

Answer

```
import java.util.Scanner;

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


      int number = sc.nextInt();


      double converted = (double) number;


      System.out.println("Original Integer: " + number);
      System.out.println("Converted Double: " + converted);
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Vishal and Arun are discussing the properties of numbers. Vishal gives Arun two integers. He asks Arun to check if the sum of these two numbers is a multiple of their product.

Can you assist Arun and determine whether the sum is a multiple of the product?

Input Format

The input consists of two space-separated integers.

Output Format

The output prints:

1. "Sum is Multiple of Product" if the sum of the two numbers is divisible by their product.
2. "Sum is Not Multiple of Product" otherwise.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1 2

Output: Sum is Not Multiple of Product

Answer

```
import java.util.Scanner;

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

      int a = sc.nextInt();
      int b = sc.nextInt();

      int sum = a + b;
      int product = a * b;

      if (sum % product == 0) {
         System.out.println("Sum is Multiple of Product");
      } else {
         System.out.println("Sum is Not Multiple of Product");
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement:

Emily has a beautiful circular garden in her backyard. She's interested in calculating two important measurements for her garden: the circumference and the area. To do this, she needs a program that can take the radius of her circular garden as input and provide the calculated circumference and area as output. The formulas she should use are as follows:

To calculate the circumference (C) of a circle, you can use the formula:

C = 2 * π * r

A = π * r^2

Where:

C represents the circumference.

A represents the area.

π (pi) is approximately 3.14159.

r is the radius of the circle.

Emily is not a programmer, and she needs your help to create a program that will make these calculations for her garden.

Input Format

The first line of input contains a single double-point number radius, representing the radius of the circle.

Output Format

The output should consist of two lines:

The first line should print the circumference of the circle rounded to 2 decimal places, followed by the unit "meters".

The second line should print the area of the circle rounded to 2 decimal places, followed by the unit "square meters".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3.0

Output: Circumference: 18.85 meters
Area: 28.27 square meters

Answer

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

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

```
        double radius = sc.nextDouble();


        double pi = 3.14159;


        double circumference = 2 * pi * radius;
        double area = pi * radius * radius;

        System.out.printf("Circumference: %.2f meters%n", circumference);
        System.out.printf("Area: %.2f square meters%n", area);
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q6

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Joey is learning about bitwise operations and is working on a project that involves extracting specific bits from integers. He needs to write a program that takes an integer and the number of bits N as input and outputs the value of the lowest N bits of the integer.

Help Joey in his project to understand and visualize how bitwise operations work in practical scenarios.

Input Format

The first line of input consists of an integer X, representing the given integer.

The second line consists of an integer N, representing the number of bits to extract.

Output Format

The output displays "Result: " followed by an integer representing the value of the lowest N bits of the given integer.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 85
2

Output: Result: 1

Answer

```
import java.util.Scanner;

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


      int x = sc.nextInt();


      int n = sc.nextInt();


      int mask = (1 << n) - 1;


      int result = x & mask;


      System.out.println("Result: " + result);
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arun is working on a project to automate the process of determining whether a student has passed or failed based on their subject marks.

He aims to create a simple program that takes positive integers as marks for five subjects from the user. If the average of the marks is greater than or equal to 50, the student has passed the exam. Otherwise, the student has failed.

Help Arun to implement the project.

Input Format

The input consists of five space-separated integers, representing the marks in five subjects.

Output Format

The first line of output prints "Average score: " followed by an integer representing the average score.

The second line prints one of the following:

1. If the condition is satisfied, print "The student has passed".
2. Otherwise, the output prints "The student has failed".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50 60 70 80 90

Output: Average score: 70
The student has passed

Answer

```
// You are using Java
import java.util.Scanner;
class main{
   public static void main(String args[]){
      Scanner s=new Scanner(System.in);
      int a,b,c,d,e,f,g;
      a=s.nextInt();
      b=s.nextInt();
      c=s.nextInt();
      d=s.nextInt();
      e=s.nextInt();
      f=(a+b+c+d+e);
      g=f/5;
      System.out.println("Average score: "+g);
      if(g>=50)
      System.out.println("The student has passed");
      else
      System.out.println("The student has failed");


   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Samantha is a diligent math student who is exploring the world of programming. She is learning Java and has recently studied conditional statements. One day, her teacher gives her an interesting problem to solve, which takes a number as input and checks whether it is a multiple of 5 or 7.

Help her complete the task.

Input Format

The input consists of a single integer N, representing the number to be checked.

Output Format

If the number is a multiple of 5 but not 7, the output prints "N is a multiple of 5".

If the number is a multiple of 7, the output prints "N is a multiple of 7".

Otherwise the output prints "N is neither multiple of 5 nor 7" where N is an entered integer.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10
Output: 10 is a multiple of 5

Answer

```
// You are using Java
import java.util.*;
class main{
    public static void main(String args[]) {
    Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        if(a%5==0){
            System.out.println(a+" is a multiple of 5");

    }else if(a%7==0){
      System.out.println(a+" is a multiple of 7");
    }else
    System.out.println(a+" is neither multiple of 5 nor 7");
    }

}
```

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

Rajalakshmi Engineering College

Scan to verify results

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John is a fitness trainer, and he wants to use the BMI calculator to assess the body mass index of his clients. He has a list of clients based on their height and weight.

John plans to write a program to quickly determine the BMI and provide a classification for each client.

If BMI is less than 18.5, the program will classify it as "Underweight"If BMI is between 18.6 and 24.9, the program will classify it as "Normal Weight"If BMI is between 25.0 and 29.9, the program will classify it as "Overweight"If BMI is 30.0 or higher, the program will classify it as "Obese"

Note: Formula to calculate BMI = weight/(height*height)

Input Format

The first line of input consists of a double value, representing the height of the person in meters.

The second line consists of a double value, representing the weight of the person in kilograms.

Output Format

The first line of output prints "BMI: " followed by a double (rounded to two decimal places) representing the calculated BMI.

The second line prints "Classification: " followed by a string indicating the BMI category (Underweight, Normal Weight, Overweight, or Obese).

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1.2
45.2

Output: BMI: 31.39
Classification: Obese

Answer

```
import java.util.Scanner;

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

        // Read inputs
        double height = sc.nextDouble();
        double weight = sc.nextDouble();

        // Calculate BMI
        double bmi = weight / (height * height);

        // Classification
        String classification;
        if (bmi < 18.5) {
            classification = "Underweight";
        } else if (bmi <= 24.9) {
            classification = "Normal Weight";
        } else if (bmi <= 29.9) {
            classification = "Overweight";
        } else {
            classification = "Obese";
        }

        // Print output, rounding BMI to 2 decimal places
        System.out.printf("BMI: %.2f Classification: %s\n", bmi, classification);

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 9

Section 1 : Coding

1. Problem Statement

Amit wants to evaluate the depreciation of his car over time to understand its current value and categorize it based on that value.

Write a program that helps him determine the current value of his car after a certain number of years of depreciation and classify it into one of three categories:

High: If the current value is greater than 10,000.Medium: If the current value is between 5,000 and 10,000, both inclusive.Low: If the current value is less than 5,000.

The depreciation rate of the car is 15% per year. The program should calculate the current value of the car after applying this depreciation over the given number of years and print the current value along with the category.

Input Format

The first line of input consists of an integer, representing the initial cost of the car.

The second line consists of an integer, representing the number of years the car has been depreciating.

Output Format

The first line of output prints a double value, representing the current value of the car, rounded off to two decimal places "Current Value: <value>".

The second line prints its category "Category: <categories>".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20000
5

Output: Current Value: 8874.11
Category: Medium

Answer

```
import java.util.Scanner;

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

      int initialCost = sc.nextInt();
      int years = sc.nextInt();

      double rate = 0.15;
      double currentValue = initialCost * Math.pow(1 - rate, years);

      currentValue = Math.round(currentValue * 100.0) / 100.0;

      String category;
      if (currentValue > 10000) {
          category = "High";
      } else if (currentValue >= 5000) {
          category = "Medium";
      } else {
          category = "Low";
      }

      System.out.printf("Current Value: %.2f\n",currentValue);
      System.out.printf("Category: %s",category);

      sc.close();
   }
}
```

Status : Partially correct ***Marks : 9/10***

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ted, the computer science enthusiast, has accepted the challenge of writing a program that checks if the number of digits in an integer matches the sum of its digits.

Guide Ted in designing and writing the code to solve this problem using a 'do-while' loop.

Input Format

The input consists of an integer N, representing the number to be checked.

Output Format

If the sum is equal to the number of digits, print "The number of digits in N matches the sum of its digits."

Else, print "The number of digits in N does not match the sum of its digits."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20
Output: The number of digits in 20 matches the sum of its digits.

Answer

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

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

      int number = N; // store original number for output
      int sumDigits = 0;
      int countDigits = 0;

      do {
         int digit = N % 10;
         sumDigits += digit;
         countDigits++;
         N = N / 10;
      } while (N > 0);

      if (sumDigits == countDigits) {
         System.out.println("The number of digits in " + number + " matches the
sum of its digits.");
      } else {
         System.out.println("The number of digits in " + number + " does not match
the sum of its digits.");
      }

      sc.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Rosh is intrigued by numerical patterns. Today, she stumbled upon a puzzle while working with arrays. She wants to compute the sum of the third-largest and second-smallest elements from a list of integers. She seeks your help to implement a program that solves this for her efficiently.

Input Format

The first line of input is an integer N, representing the size of the array.

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

Output Format

The output displays a single integer representing the sum of the third-largest and second-smallest elements in the array.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10
10 20 30 40 50 60 70 80 90 100

Output: 100

Answer

```
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();
        }


        Arrays.sort(arr);


        int thirdLargest = arr[N - 3];
        int secondSmallest = arr[1];


        System.out.println(thirdLargest + secondSmallest);

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Monica is interested in finding a treasure but the key to opening is to get the sum of the main diagonal elements and secondary diagonal elements.

Write a program to help Monica find the diagonal sum of a square 2D array.

Note: The main diagonal of the array consists of the elements traversing from the top-left corner to the bottom-right corner. The secondary diagonal includes elements from the top-right corner to the bottom-left corner.

Input Format

The first line of input consists of an integer N, representing the number of rows and columns.

The following N lines consist of N space-separated integers, representing the 2D array elements.

Output Format

The first line of output prints "Sum of the main diagonal: " followed by an integer, representing the sum of the main diagonal.

The second line prints "Sum of the secondary diagonal: " followed by an integer, representing the sum of the secondary diagonal.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
1 2 3
4 5 6
7 8 9

Output: Sum of the main diagonal: 15
Sum of the secondary diagonal: 15

Answer

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

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


        int N = sc.nextInt();

        int[][] matrix = new int[N][N];

        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                matrix[i][j] = sc.nextInt();
            }
        }
        int mainDiagonalSum = 0;
        int secondaryDiagonalSum = 0;


        for (int i = 0; i < N; i++) {
            mainDiagonalSum += matrix[i][i];
            secondaryDiagonalSum += matrix[i][N - 1 - i];
        }


        System.out.println("Sum of the main diagonal: " + mainDiagonalSum);
        System.out.println("Sum of the secondary diagonal: " +
secondaryDiagonalSum);

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sesha is developing a weather monitoring system for a region with multiple weather stations. Each weather station collects temperature data hourly and stores it in a 2D array.

Write a program that can add the temperature data from two different weather stations to create a combined temperature record for the region.

Input Format

The first line of input consists of two space-separated integers N and M, representing the number of rows and columns of the matrices, respectively.

The next N lines consist of M space-separated integers, representing the values of the first matrix.

The following N lines consist of M space-separated integers, representing the values of the second matrix.

Output Format

The output prints the addition of the two matrices in N rows and M columns, representing the combined temperature record.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3 3
1 2 3
4 5 6
7 8 9
1 1 1
2 2 2
3 3 3

Output: 2 3 4
6 7 8
10 11 12

Answer

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

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

      // Read dimensions N (rows) and M (columns)
      int N = sc.nextInt();
      int M = sc.nextInt();

      int[][] matrix1 = new int[N][M];
      int[][] matrix2 = new int[N][M];
      int[][] result = new int[N][M];

      // Read first matrix
      for (int i = 0; i < N; i++) {
          for (int j = 0; j < M; j++) {
              matrix1[i][j] = sc.nextInt();
          }
      }

      // Read second matrix
      for (int i = 0; i < N; i++) {
          for (int j = 0; j < M; j++) {
              matrix2[i][j] = sc.nextInt();
          }
      }

      // Add the two matrices
      for (int i = 0; i < N; i++) {
          for (int j = 0; j < M; j++) {
              result[i][j] = matrix1[i][j] + matrix2[i][j];
          }
      }

      // Print the result matrix in required format
      for (int i = 0; i < N; i++) {
          for (int j = 0; j < M; j++) {
              System.out.print(result[i][j] + " ");
          }
      }
      System.out.println();

      sc.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sharon is creating a program that finds the first repeated element in an integer array. The program should efficiently identify the first element that appears more than once in the given array. If no such element is found, it should appropriately display a message.

Help Sharon to complete the program.

Input Format

The first line of input consists of an integer n, representing the number of elements in the array.

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

Output Format

If a repeated element is found, print the first element that appears more than once.

If no repeated element is found, print "No repeated element found in the array".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 8
12 21 13 14 21 36 47 21

Output: 21

Answer

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

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

        // Read number of elements
        int n = sc.nextInt();
        int[] arr = new int[n];

        // Read array elements
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        // Use a HashSet to track seen elements
        Set<Integer> seen = new HashSet<>();
        int firstRepeated = -1;

        for (int num : arr) {
            if (seen.contains(num)) {
                firstRepeated = num;
                break;
            } else {
                seen.add(num);
            }
        }

        if (firstRepeated == -1) {
            System.out.println("No repeated element found in the array");
        } else {
            System.out.println(firstRepeated);
        }

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 4_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 10

Section 1 : MCQ

1. Predict the output for the following code:

```
class Main {
   public static void main(String args[]) {
      StringBuffer sb = new StringBuffer("I Java!");
      sb.insert(5, "like ");
      System.out.println(sb);
   }
}
```

Answer

I Javlike a!

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

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

```
public class Main {
    public static void main(String[] args) {
      String str = "1234.34";
      int a = Integer.parseInt(str);
      System.out.println(a);
    }
}
```

Answer

NumberFormatException

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

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

```
class Main {
    public static void main(String args[]) {
        String s1 = "Hello i love java";
        String s2 = new String(s1);
        System.out.println((s1 == s2) + " " + s1.equals(s2));
    }
}
```

Answer

false false

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

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

```
class Main {
    public static void main(String[] args) {
        String s1 = "EDUCATION";
        String s2 = new String("EDUCATION");
        String s3 = "EDUCATION";
        if (s1 == s2) {
            System.out.println("s1 and s2 equal");
        }
        else {
            System.out.println("s1 and s2 not equal");
        }
        if (s1 == s3) {
            System.out.println("s1 and s3 equal");
        }
        else {
            System.out.println("s1 and s3 not equal");
        }
    }
}
```

Answer

s1 and s2 not equals1 and s3 equal

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

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

```
class Main {
    public static void main(String args[])  {
        String name="Work Hard";
        name.concat("Success");
        System.out.println(name);
    }
}
```

Answer

Work Hard

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

6. Predict the output for the following code:

```
public class Main {
    public static void main(String[] args) {
        float a = 10.0f;
        String temp = Float.toString(a);
    System.out.println(temp);
  }
}
```

Answer

10.0

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

7. Predict the output for the following code.

```
class Main {
  public static void main(String[] fruits) {
    String fruit1 = new String("apple");
    String fruit2 = new String("orange");
    String fruit3 = new String("pear");
    fruit3 = fruit1;
    fruit2 = fruit3;
    fruit1 = fruit2;
    System.out.println(fruit1);
    System.out.println(fruit2);
    System.out.println(fruit3);
  }
}
```

Answer

pearpearpear

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

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

```
class Main {
  public static void main(String args[]) {
    StringBuffer sb = new StringBuffer("Hello");
    System.out.println("buffer = " + sb);
    System.out.println("length = " + sb.length());
    System.out.println("capacity = " + sb.capacity());
  }
}
```

Answer

buffer = Hellolength = 5capacity = 23

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

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

```
class Main {
  public static void main(String[] args) {
    String greet = "Welcome\n";
    System.out.print("String: " + greet);
    int length = greet.length();
    System.out.print("Length: " + length);
  }
}
```

Answer

String: WelcomeLength: 8

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

10. Predict the output for the following code.

```
public class Main {
    public static void main(String[] args) {
      String a = "java";
      char temp = a.charAt(1);
      System.out.println(temp);
  }
}
```

Answer

a

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

11. What is the output of the following code?

```
class Main
{
   public static void main(String args[])
   {
      StringBuffer c = new StringBuffer("Hello");
      c.delete(0,2);
      System.out.println(c);
   }
}
```

Answer

llo

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

12. What will be the output for the following code?

```
class Main {
   public static void main(String[] args) {
   String languages[] = { "C", "C++", "Java", "Python", "Ruby"};
      for (String sample: languages) {
         System.out.println(sample);
      }
   }
}
```

Answer

C, C++, Java, Python, Ruby

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

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

```
class Main {
   public static void main(String args[]) {
      char c[] = {'j', 'a', 'v', 'a'};
      String s1 = new String(c);
      String s2 = new String(s1);
```

```
        System.out.println(s1);
        System.out.println(s2);
    }
}
```

Answer

javajava

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

14. What will be the output of the following program?

```
class Main {
    public static void main(String[] args) {
        String s = new String("5");
        System.out.println(1 + 1111 + s + 1 + 1010);
    }
}
```

Answer

1112511010

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

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

```
class Main {
    public static void main(String args[])
    {
        StringBuffer sb = new StringBuffer("Hello");
        System.out.println("buffer before = " + sb);
        System.out.println("charAt(1) before = " + sb.charAt(1));
        sb.setCharAt(1, 'i');
        sb.setLength(2);
        System.out.println("buffer after = " + sb);
        System.out.println("charAt(1) after = " + sb.charAt(1));
    }
}
```

Answer

buffer before = HellocharAt(1) before = lbuffer after = HicharAt(1) after = i

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 6.5

Section 1 : Coding

1. Problem Statement

Anu is developing a tool for a conference registration system. Participants submit keywords related to their fields of interest. The organizer wants to sort these keywords alphabetically to generate tags for session grouping.

Write a program that accepts at least five keywords as input arguments and outputs them in sorted alphabetical order.

Input Format

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

The second line of input contains n space-separated keywords (string).

Output Format

The output prints n space separated strings representing the sorted keyword in alphabetical order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
Blockchain Cloud AI Data Cybersecurity
Output: AI Blockchain Cloud Cybersecurity Data

Answer

```
// You are using Java
import java.util.Arrays;
import java.util.Scanner;

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

        int n = sc.nextInt();
        sc.nextLine();  // consume newline

        String[] keywords = sc.nextLine().split(" ");


        Arrays.sort(keywords, String.CASE_INSENSITIVE_ORDER);


        for (int i = 0; i < n; i++) {
            System.out.print(keywords[i]);
            if (i != n - 1) {
                System.out.print(" ");
            }
        }
        System.out.println();

        sc.close();
    }
}
```

Status : Partially correct *Marks : 6.5/10*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bechan Chacha is seeking help to filter out valid mobile numbers from a list provided by his crush. He can only pick his crush's number if the list contains valid mobile numbers.

A mobile number is considered valid if:

It has exactly 10 digits.It consists only of numeric values (0–9).It does not begin with zero.

Your task is to determine whether each mobile number in the list is valid or not.

Input Format

The first line contains an integer T, representing the number of mobile numbers to check.

The next T lines each contain a string S, representing a mobile number.

Output Format

For each mobile number S, the output print "YES" if it is valid.

Otherwise, print "NO".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
9876543210

Output: YES

Answer

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

public class  Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = sc.nextInt();
        for (int i = 0; i < T; i++) {
            String s = sc.next();
            if (s.length() == 10 && s.matches("[1-9][0-9]{9}")) {
                System.out.println("YES");
            } else {
                System.out.println("NO");
            }
        }
        sc.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arjun is learning how to filter words from a sentence based on grammar rules. He wants to identify the valid words in a sentence.

A word is considered valid if it satisfies all these conditions:

The word contains only alphabets (a–z, A–Z).The word length is at least 2 characters.The word should not contain digits or special characters.

Your task is to read a sentence and print all the valid words in it.

Input Format

The input contains a single line containing a sentence S.

Output Format

The output prints all the valid words separated by spaces.

If no valid word exists, print "No valid words."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Hello world1 123 ab" @#$ Hi
Output: Hello Hi

Answer

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

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String[] words = sc.nextLine().split(" ");
      StringBuilder sb = new StringBuilder();
      for (String word : words) {
         if (word.length() >= 2 && word.matches("[a-zA-Z]+")) {
            sb.append(word).append(" ");
         }
      }
      if (sb.length() > 0) {
         System.out.println(sb.toString());
      } else {
         System.out.println("No valid words.");
      }
      sc.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a secure banking system, customers are required to create PIN codes for accessing their accounts. The bank wants to validate these PIN codes before accepting them.

A PIN code is considered valid if:

It consists of exactly 4 digits.All characters must be numeric (0–9).It cannot contain all identical digits (e.g., 1111 is invalid).

Your task is to determine whether each PIN code in the list is valid or not.

Input Format

The first line of input contains an integer T, representing the number of PIN codes to check.

The next T lines each contain a string S, representing a PIN code.

Output Format

For each PIN code S, the output print "YES" if it is valid.

Otherwise, the output print "NO".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234

Output: YES

Answer

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = sc.nextInt();
        for (int i = 0; i < T; i++) {
            String pin = sc.next();
            if (pin.matches("\\d{4}") && !allSame(pin)) {
                System.out.println("YES");
            } else {
                System.out.println("NO");
            }
        }
        sc.close();
    }

    static boolean allSame(String s) {
        char first = s.charAt(0);
        for (int i = 1; i < s.length(); i++) {
            if (s.charAt(i) != first) return false;
        }
        return true;
    }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are working as a developer for CityBank, which wants to build a basic account management system.

Each customer at the bank has:

An Account Number (integer)A Customer Name (string)An Initial Balance (double)

The bank allows two types of transactions:

Deposit – increases the balance.Withdrawal – decreases the balance only if enough funds are available.

If the withdrawal amount is greater than the balance, the withdrawal should not happen, and the balance should remain the same.

You are required to implement this system using:

A class with attributes for account details.A constructor to initialize account details.Setter methods to update details if needed.Getter methods to retrieve details.Objects of the class to represent customers.

Finally, display each customer's account details after all transactions.

Input Format

The first line of input contains an integer N, representing the number of customers.

For each customer:

- The next line contains the account number (integer).
- The following line contains the customer name (string).
- The next line contains the initial balance (double).
- The next line contains the deposit amount (double).
- The next line contains the withdrawal amount (double).

Output Format

For each customer, print the details in the following format:

1. Account Number: <account_number>
2. Customer Name: <customer_name>
3. Final Balance: <final_balance> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234
Rahul Sharma
5000
2000
3000

Output: Account Number: 1234
Customer Name: Rahul Sharma

```
Final Balance: 4000.0
```

Answer

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

class BankAccount {
    private int accountNumber;
    private String customerName;
    private double balance;

    // Constructor to initialize account details
    public BankAccount(int accountNumber, String customerName, double
initialBalance) {
        this.accountNumber = accountNumber;
        this.customerName = customerName;
        this.balance = initialBalance;
    }

    // Getter methods
    public int getAccountNumber() {
        return accountNumber;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getBalance() {
        return balance;
    }

    // Method to deposit amount
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
        }
    }

    // Method to withdraw amount
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
        }
    }

    // Method to display account details
    public void displayAccountDetails() {
        System.out.printf("Account Number: %d\n", accountNumber);
        System.out.printf("Customer Name: %s\n", customerName);
        System.out.printf("Final Balance: %.1f\n", balance);
    }
}

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

        // Input number of customers
        int N = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        // Process each customer
        for (int i = 0; i < N; i++) {
            // Input account details
            int accountNumber = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            String customerName = scanner.nextLine();
            double initialBalance = scanner.nextDouble();
            double depositAmount = scanner.nextDouble();
            double withdrawalAmount = scanner.nextDouble();


            // Create BankAccount object
            BankAccount account = new BankAccount(accountNumber,
customerName, initialBalance);

            // Perform transactions
            account.deposit(depositAmount);
            account.withdraw(withdrawalAmount);

            // Display account details
            account.displayAccountDetails();
        }
        scanner.close();
    }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Neha is working as a developer for CityElectricity Board, which wants to build a household electricity billing system.

Each customer’s electricity account has:

A Customer ID (integer)A Customer Name (string)Units Consumed (double)

The electricity bill is calculated based on these rules:

For the first 100 units 5 units charge per unitFor the next 100 units (101–200) 7 units charge per unitFor units above 200 10 units charge per unitIf the total bill exceeds 2000 units, a 5% discount is applied on the final bill.

Neha has been asked to implement this system using:

A class with attributes for customer details.A constructor to initialize customer details.Setter methods to update details if needed.Getter methods to retrieve details.Objects of the class to represent customers.

Finally, display each customer’s details and final bill amount.

Input Format

The first line of input contains an integer N, representing the number of customers.

For each customer:

- The next line contains the Customer ID (integer).
- The following line contains the Customer Name (string).
- The next line contains the Units Consumed (double).

Output Format

For each customer, print the details in the following format:

Customer ID: <customer_id>

Customer Name: <customer_name>

Final Bill: <final_bill> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1001
Ravi Kumar
80

Output: Customer ID: 1001
Customer Name: Ravi Kumar
Final Bill: 400.0

Answer

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

```
class Customer {
    private int customerId;
    private String customerName;
    private double unitsConsumed;

    public Customer(int customerId, String customerName, double
unitsConsumed) {
        this.customerId = customerId;
        this.customerName = customerName;
        this.unitsConsumed = unitsConsumed;
    }

    public int getCustomerId() {
        return customerId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getUnitsConsumed() {
        return unitsConsumed;
    }

    public double calculateBill() {
        double bill = 0.0;
        double units = unitsConsumed;

        if (units <= 100) {
            bill = units * 5;
        } else if (units <= 200) {
            bill = 100 * 5 + (units - 100) * 7;
        } else {
            bill = 100 * 5 + 100 * 7 + (units - 200) * 10;
        }

        if (bill > 2000) {
            bill = bill - (bill * 0.05);
        }

        return bill;
    }
}

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

        for (int i = 0; i < n; i++) {
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            double units = Double.parseDouble(sc.nextLine());

            Customer customer = new Customer(id, name, units);
            customers.add(customer);
        }

        for (Customer c : customers) {
            System.out.printf("Customer ID: %d\n", c.getCustomerId());
            System.out.printf("Customer Name: %s\n", c.getCustomerName());
            System.out.printf("Final Bill: %.1f\n", c.calculateBill());
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are working as a developer for CityCab, a taxi service company that wants to build a ride fare management system.

Each customer booking has:

A Booking ID (integer)A Customer Name (string)A Distance Travelled in km (double)

The fare calculation rules are:

Base Fare = 50 units (flat charge for every ride).Per km charge = 10 units/km.If the distance is greater than 20 km, a 10% discount is applied on the total fare.

You are required to implement this system using:

A class with attributes for booking details.A constructor to initialize booking details.Setter methods to update details if needed.Getter methods to retrieve details.Objects of the class to represent customer rides.

Finally, display each booking's details and final fare.

Input Format

The first line of input contains an integer N, representing the number of bookings.

For each booking:

- The next line contains the booking ID (integer).
- The following line contains the customer's name (string).
- The next line contains the distance travelled (double).

Output Format

For each booking, print the details in the following format:

1. Booking ID: <booking_id>
2. Customer Name: <customer_name>
3. Final Fare: <final_fare> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234
Rahul Sharma
15

Output: Booking ID: 1234
Customer Name: Rahul Sharma
Final Fare: 200.0

Answer

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

```
class Booking {
   private int bookingId;
   private String customerName;
   private double distance;

   // Constructor
   public Booking(int bookingId, String customerName, double distance) {
      this.bookingId = bookingId;
      this.customerName = customerName;
      this.distance = distance;
   }

   // Setter methods
   public void setBookingId(int bookingId) {
      this.bookingId = bookingId;
   }

   public void setCustomerName(String customerName) {
      this.customerName = customerName;
   }

   public void setDistance(double distance) {
      this.distance = distance;
   }

   // Getter methods
   public int getBookingId() {
      return bookingId;
   }

   public String getCustomerName() {
      return customerName;
   }

   public double getDistance() {
      return distance;
   }

   // Method to calculate final fare
   public double calculateFare() {
      double baseFare = 50;
      double perKmCharge = 10;
```

```
        double totalFare = baseFare + (distance * perKmCharge);

        if (distance > 20) {
            totalFare = totalFare * 0.9; // 10% discount
        }

        return Math.round(totalFare * 10) / 10.0; // round to 1 decimal place
    }
}

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

        int N = Integer.parseInt(sc.nextLine().trim());

        Booking[] bookings = new Booking[N];

        for (int i = 0; i < N; i++) {
            int bookingId = Integer.parseInt(sc.nextLine().trim());
            String customerName = sc.nextLine().trim();
            double distance = Double.parseDouble(sc.nextLine().trim());

            bookings[i] = new Booking(bookingId, customerName, distance);
        }

        for (Booking booking : bookings) {
            System.out.println("Booking ID: " + booking.getBookingId());
            System.out.println("Customer Name: " + booking.getCustomerName());
            System.out.println("Final Fare: " + booking.calculateFare());
        }

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ram is working as a developer for BrightEdu Coaching Center, which wants to build a student fee management system.

Each student’s enrollment has:

An Enrollment ID (integer)A Student Name (string)The Number of Subjects (integer)

The fee calculation rules are:

Registration Fee = 1000 units (flat for every student).Per Subject Fee = 800 units.If the student enrolls in more than 5 subjects, a 20% scholarship (discount) is applied on the total fee.

Ram has been asked to implement this system using:

A class with attributes for student details.A constructor to initialize student details.Setter methods to update details if needed.Getter methods to retrieve details.Objects of the class to represent student enrollments.

Finally, display each student’s details and final fee.

Input Format

The first line of input contains an integer N, representing the number of students.

For each student:

- The next line contains the Enrollment ID (integer).
- The following line contains the student’s name (string).
- The next line contains the Number of subjects (integer).

Output Format

For each student, print the details in the following format:

- Enrollment ID: <enrollment_id>
- Student Name: <student_name>
- Final Fee: <final_fee> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234
Ravi Kumar
3

Output: Enrollment ID: 1234
Student Name: Ravi Kumar
Final Fee: 3400.0

Answer

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

class Student {
```

```
private int enrollmentId;
private String studentName;
private int numberOfSubjects;

public Student(int enrollmentId, String studentName, int numberOfSubjects) {
    this.enrollmentId = enrollmentId;
    this.studentName = studentName;
    this.numberOfSubjects = numberOfSubjects;
}

public void setEnrollmentId(int enrollmentId) {
    this.enrollmentId = enrollmentId;
}

public void setStudentName(String studentName) {
    this.studentName = studentName;
}

public void setNumberOfSubjects(int numberOfSubjects) {
    this.numberOfSubjects = numberOfSubjects;
}

public int getEnrollmentId() {
    return enrollmentId;
}

public String getStudentName() {
    return studentName;
}

public int getNumberOfSubjects() {
    return numberOfSubjects;
}

public double calculateFee() {
    double registrationFee = 1000;
    double perSubjectFee = 800;
    double totalFee = registrationFee + (numberOfSubjects * perSubjectFee);

    if (numberOfSubjects > 5) {
        totalFee = totalFee * 0.8;
    }
    return Math.round(totalFee * 10) / 10.0;
  }
}

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

    int N = Integer.parseInt(sc.nextLine().trim());

    for (int i = 0; i < N; i++) {
      int enrollmentId = Integer.parseInt(sc.nextLine().trim());
      String studentName = sc.nextLine().trim();
      int numberOfSubjects = Integer.parseInt(sc.nextLine().trim());

      Student student = new Student(enrollmentId, studentName,
numberOfSubjects);

      System.out.println("Enrollment ID: " + student.getEnrollmentId());
      System.out.println("Student Name: " + student.getStudentName());
      System.out.println("Final Fee: " + student.calculateFee());
    }

    sc.close();
  }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Elsa subscribes to a premium service with a base monthly cost, a service tax and an extra feature cost. Assist her in writing an inheritance program that takes input for these values and calculates the total monthly cost.

Refer to the below class diagram:

Input Format

The first line of input consists of a double value, representing the base monthly cost.

The second line consists of a double value, representing the service tax.

The third line consists of a double value, representing the extra feature cost.

Output Format

The output prints "Rs. X" where X is a double value, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10.0
2.5
5.0
Output: Rs. 17.50

Answer

```
import java.util.Scanner;

// You are using Java


// Base class representing a general subscription
class Subscription {
   protected double baseMonthlyCost;

   public Subscription(double baseMonthlyCost) {
      this.baseMonthlyCost = baseMonthlyCost;
   }
}

// Derived class representing a premium subscription
class PremiumSubscription extends Subscription {
   private double serviceTax;
   private double extraFeatureCost;

   public PremiumSubscription(double baseMonthlyCost, double serviceTax,
double extraFeatureCost) {
      super(baseMonthlyCost); // Call to the parent constructor
      this.serviceTax = serviceTax;
      this.extraFeatureCost = extraFeatureCost;
   }
   public double calculateMonthlyCost() {
      return baseMonthlyCost + serviceTax + extraFeatureCost;
   }
}

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

      double baseMonthlyCost = scanner.nextDouble();
      double serviceTax = scanner.nextDouble();
      double extraFeatureCost = scanner.nextDouble();

      PremiumSubscription premiumSubscription = new
PremiumSubscription(baseMonthlyCost, serviceTax, extraFeatureCost);

      double totalMonthlyCost = premiumSubscription.calculateMonthlyCost();

      System.out.printf("Rs. %.2f%n", totalMonthlyCost);

      scanner.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Alice is managing an online store and wants to implement a program using inheritance to calculate the selling price of products after applying discounts.

Guide her by following the instructions:

Create a base class called Product with a public double attribute price.Create a subclass called DiscountedProduct, which extends Product and includes a private double attribute discount rate. This subclass has a method called calculateSellingPrice() to determine the final selling price after applying the discount.

Formula: Discounted selling price = price * (1 - discount rate)

Input Format

The first line of input consists of a double value p, the initial price of the product.

The second line consists of a double value d, the discount rate.

Output Format

The output prints "Rs. X", where X is a double value, representing the calculated discounted selling price, rounded off to two decimal places.

If the discount rate is greater than 1, print "Not applicable".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00
0.20

Output: Rs. 40.00

Answer

```
import java.util.Scanner;

import java.util.Scanner;

class Product {
   public double price;

   public Product(double price) {
      this.price = price;
   }
}

class DiscountedProduct extends Product {
   private double discountRate;

   public DiscountedProduct(double price, double discountRate) {
      super(price);
      this.discountRate = discountRate;
   }

   public double calculateSellingPrice() {
        if (discountRate > 1.00) {
            return -1; // Indicate that the discount rate is not applicable
        } else {
            return price * (1 - discountRate);
        }
    }
}


class ProductPricing {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double initialPrice = scanner.nextDouble();
        double discountRate = scanner.nextDouble();
        DiscountedProduct discountedProduct = new
DiscountedProduct(initialPrice, discountRate);
        double sellingPrice = discountedProduct.calculateSellingPrice();

        if (sellingPrice >= 0) {
            System.out.printf("Rs. %.2f%n", sellingPrice);
        } else {
            System.out.println("Not applicable");
        }
        scanner.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Preethi is working on a project to automate sales tax calculations for items in a store. She wants to create a program that takes the price of an item and the sales tax rate as input and calculates the final price of the item after applying the sales tax.

Write a program using the class SalesTaxCalculator, which contains an overloaded method named calculateFinalPrice to handle both integer and double inputs. The program should also include a Main class that takes user input, calls the appropriate method from SalesTaxCalculator, and prints the final price of the item.

Formula Used: Final price = price + ((price * sales tax rate) / 100)

Input Format

The first line of input consists of an integer price (the price of the item for integer inputs).

The second line of input consists of an integer taxRate (the sales tax rate for integer inputs).

The third line of input consists of a double price (the price of the item for double inputs).

The fourth line of input consists of a double taxRate (the sales tax rate for double inputs).

Output Format

The first line of output prints an integer, representing the final price of the item after applying the sales tax for integer inputs (a and b).

The second line prints a double value, representing the final price of the item after applying the sales tax for double-value inputs (m and n), rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 100
10
100.0
5.0
Output: 110
105.00
```

Answer

```
import java.util.Scanner;

// You are using Java
import java.util.Scanner;

class SalesTaxCalculator {
  // Method to calculate final price for integer inputs
  public static int calculateFinalPrice(int price, int taxRate) {
      return price + (price * taxRate) / 100;
  }

  // Method to calculate final price for double inputs
  public static double calculateFinalPrice(double price, double taxRate) {
      return price + (price * taxRate) / 100;
  }
}


class Main {
  public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      int intPrice = scanner.nextInt();
      int intTaxRate = scanner.nextInt();
      double doublePrice = scanner.nextDouble();
      double doubleTaxRate = scanner.nextDouble();

      int finalPriceInt = SalesTaxCalculator.calculateFinalPrice(intPrice,
intTaxRate);
      double finalPriceDouble =
SalesTaxCalculator.calculateFinalPrice(doublePrice, doubleTaxRate);

      System.out.println(finalPriceInt);
      System.out.format("%.2f", finalPriceDouble);
  }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Mr.Kapoor wants to create a program to calculate the volume of a Cuboid and a Cube using method overriding.

Implements a base class Cuboid with attributes for length, width, and height. Include a method calculateVolume() that computes the volume of the cuboid.

Extends the base class with a subclass Cube representing a cube, where all sides are equal. Override the calculateVolume() method in the Cube class to compute the volume of the cube.

The program should take user input for the dimensions of the cuboid and the side length of the cube and display the calculated volumes with two decimal places.

Input Format

The first line of input consists of 3 space-separated double values, representing the cuboid length, width, and height, respectively.

The second line consists of a double value, representing the side length of the cube.

Output Format

The first line of output prints the volume of the cuboid, rounded off to two decimal places.

The second line prints the volume of the cube, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 60.0 60.0 60.0
50.0

Output: Volume of Cuboid: 216000.00
Volume of Cube: 125000.00

Answer

```
import java.util.Scanner;

// You are using Java
import java.util.Scanner;

// Base class Cuboid
class Cuboid {
   protected double length, width, height;

   // Constructor to initialize dimensions
   public Cuboid(double length, double width, double height) {
      this.length = length;
      this.width = width;
      this.height = height;
   }
```

```
    // Method to calculate volume of cuboid
    public double calculateVolume() {
        return length * width * height;
    }
}

// Subclass Cube
class Cube extends Cuboid {

    // Constructor to initialize side length
    public Cube(double side) {
        super(side, side, side);
    }

    // Overridden method to calculate volume of cube
    @Override
    public double calculateVolume() {
        return length * length * length;
    }
}


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

        double cuboidLength = scanner.nextDouble();
        double cuboidWidth = scanner.nextDouble();
        double cuboidHeight = scanner.nextDouble();

        // Regular object instantiation for Cuboid
        Cuboid cuboid = new Cuboid(cuboidLength, cuboidWidth, cuboidHeight);
        System.out.printf("Volume of Cuboid: %.2f\n", cuboid.calculateVolume());

        double cubeSide = scanner.nextDouble();

        // Upcasting - Using superclass reference for subclass object (DMD)
        Cuboid cube = new Cube(cubeSide); // Upcasting
        System.out.printf("Volume of Cube: %.2f", cube.calculateVolume()); // Calls
Cube's method dynamically

        scanner.close();
    }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement:

Rajiv is analyzing the energy consumption in his household and wants to calculate the total cost based on the daily energy usage. He is given the rate per unit of electricity and the energy consumed for multiple days. To structure this calculation efficiently, he decides to use an interface-based approach.

Implement an interface CostCalculator with the necessary methods to retrieve energy details and compute the cost. The calculations should be handled in the EnergyConsumptionTracker class, while the EnergyConsumptionApp class should only handle input and output.

Formula

Energy Cost for one day = Energy Consumed per day * Rate Per Unit

Input Format

The first line of input consists of the rate per unit as an 'R' (a double value).

The second line of input consists of the number of days 'N' (an integer).

The third line of input consists of the daily energy consumption values for each day 'D" (double values), separated by space.

Output Format

The first line of the output prints: "Day-wise Energy Cost:"

The next N lines of the output print the day-wise energy costs(double type) and the total energy cost (double type) in Indian Rupees in the following format: "Day [day_number]: Rs. [energy_cost]"

The last line of the output prints: "Total Energy Cost: Rs. [total_cost]"

Note: energy_cost and total_cost are rounded off to two decimal points

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 0.01
3
10.0 20.0 30.0
Output: Day-wise Energy Cost:
Day 1: Rs. 0.10
Day 2: Rs. 0.20
Day 3: Rs. 0.30
Total Energy Cost: Rs. 0.60
```

Answer

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

```
import java.util.Scanner;

// Interface definition
interface CostCalculator {
    void getEnergyDetails(Scanner scanner);
    void calculateAndDisplayCost();
}


class EnergyConsumptionTracker implements CostCalculator {
    private double ratePerUnit;
    private int numDays;
    private double[] dailyConsumption;

    public EnergyConsumptionTracker(double ratePerUnit, int numDays) {
        this.ratePerUnit = ratePerUnit;
        this.numDays = numDays;
        this.dailyConsumption = new double[numDays];
    }

    // Read daily energy consumption values
    public void getEnergyDetails(Scanner scanner) {
        for (int i = 0; i < numDays; i++) {
            dailyConsumption[i] = scanner.nextDouble();
        }
    }

    // Calculate and print day-wise and total cost
    public void calculateAndDisplayCost() {
        System.out.println("Day-wise Energy Cost:");
        double totalCost = 0.0;
        for (int i = 0; i < numDays; i++) {
            double cost = dailyConsumption[i] * ratePerUnit;
            totalCost += cost;
            System.out.printf("Day %d: Rs. %.2f%n", i + 1, cost);
        }
        System.out.printf("Total Energy Cost: Rs. %.2f%n", totalCost);
    }
}

// Main class handling I/O
class EnergyConsumptionApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double ratePerUnit = scanner.nextDouble();
        int numDays = scanner.nextInt();

        CostCalculator tracker = new EnergyConsumptionTracker(ratePerUnit,
numDays);

        tracker.getEnergyDetails(scanner);
        tracker.calculateAndDisplayCost();

        scanner.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Jaheer is working on a health monitoring system to help individuals calculate their Body Mass Index (BMI). He has implemented a basic BMI calculator and an interface called HealthCalculator. It should have a method called calculateBMI.

You are tasked with creating a program that takes weight and height as input, calculates the BMI using the BMICalculator class, and displays the result. If the height or weight is less than or equal to zero, then return -1.

Formula: BMI = weight / (height * height)

Input Format

The first line of input consists of a double value W, the person's weight in kilograms.

The second line consists of a double value H, the height of the person in meters.

Output Format

The output displays "BMI: " followed by a double value, representing the calculated BMI, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 70.0
1.75
Output: BMI: 22.86

Answer

```
import java.util.Scanner;

import java.util.Scanner;

// Interface definition
interface HealthCalculator {
   double calculateBMI(double weight, double height);
}

// BMICalculator implementing HealthCalculator
class BMICalculator implements HealthCalculator {
   public double calculateBMI(double weight, double height) {
      if (weight <= 0 || height <= 0) {
         return -1;
      }
      return weight / (height * height);
   }
}

// Main class with input/output (given)

class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
```

```
        double weight = scanner.nextDouble();
        double height = scanner.nextDouble();

        BMICalculator bmiCalculator = new BMICalculator();

        double bmi = bmiCalculator.calculateBMI(weight, height);

        System.out.printf("BMI: %.2f\n", bmi);


        scanner.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A financial analyst, Alex, needs a program to calculate simple interest for various financial transactions. He requires a straightforward tool that takes in the principal amount, interest rate, and time in years and computes the interest.

The formula to be used is: Interest = Principal × Rate × Time / 100

Implement this functionality using the InterestCalculator interface and the SimpleInterestCalculator class.

Input Format

The first line of input consists of the principal amount P as a double value.

The second line of input consists of the annual interest rate r as a double value.

The third line of input consists of the number of years t as a positive integer, which is an integer value.

Output Format

The output displays the calculated simple interest in the following format: "Simple Interest: [interest_value]", Here, [interest_value] should be replaced with the actual interest value calculated by the program.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1000.00
5.00
2

Output: Simple Interest: 100.0

Answer

```
import java.util.Scanner;

// You are using Java
import java.util.Scanner;

// Interface definition
interface InterestCalculator {
   double simpleInterest(double principal, double rate, int time);
}


class SimpleInterestCalculator implements InterestCalculator {
   @Override
   public double simpleInterest(double principal, double rate, int time) {
      return (principal * rate * time) / 100.0;
   }
}

// Main class with input/output handling
```

```
class Main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    double principal = scanner.nextDouble();

    double rate = scanner.nextDouble();

    int time = scanner.nextInt();

    InterestCalculator calculator = new SimpleInterestCalculator();

    double interest = calculator.simpleInterest(principal, rate, time);

    System.out.println("Simple Interest: " + interest);

  }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A financial analyst, Alex, needs a program to calculate simple interest for various financial transactions. He requires a straightforward tool that takes in the principal amount, interest rate, and time in years and computes the interest.

The formula to be used is: Interest = Principal × Rate × Time / 100

Implement this functionality using the InterestCalculator interface and the SimpleInterestCalculator class.

Input Format

The first line of input consists of the principal amount P as a double value.

The second line of input consists of the annual interest rate r as a double value.

The third line of input consists of the number of years t as a positive integer, which is an integer value.

Output Format

The output displays the calculated simple interest in the following format: "Simple Interest: [interest_value]", Here, [interest_value] should be replaced with the actual interest value calculated by the program.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1000.00
5.00
2

Output: Simple Interest: 100.0

Answer

```
import java.util.Scanner;

// You are using Java
import java.util.Scanner;

// Interface definition
interface InterestCalculator {
   double simpleInterest(double principal, double rate, int time);
}


class SimpleInterestCalculator implements InterestCalculator {
   @Override
   public double simpleInterest(double principal, double rate, int time) {
      return (principal * rate * time) / 100.0;
   }
}

// Main class with input/output handling
```

```
class Main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    double principal = scanner.nextDouble();

    double rate = scanner.nextDouble();

    int time = scanner.nextInt();

    InterestCalculator calculator = new SimpleInterestCalculator();

    double interest = calculator.simpleInterest(principal, rate, time);

    System.out.println("Simple Interest: " + interest);

  }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Write a program to validate the email address and display suitable exceptions if there is any mistake.

Create 3 custom exception classes as below

DotExceptionAtTheRateExceptionDomainException

A typical email address should have a " . " character, and a "@" character, and also the domain name should be valid. Valid domain names for practice be 'in', 'com', 'net', or 'biz'.

Display Invalid Dot usage, Invalid @ usage, or Invalid Domain message based on email id.

Get the email address from the user, validate the email by checking the above-mentioned criteria, and print the validity status of the input email address.

Input Format

The first line of input contains the email to be validated.

Output Format

The output prints a Valid email address or an Invalid email address along with the suitable exception

If email ends with . or contains not exactly one . after @, it throws:

DotException: Invalid Dot usage

Invalid email address

If @ appears not exactly once, it throws:

AtTheRateException: Invalid @ usage

Invalid email address

If the part after the last dot is not among accepted domains:

DomainException: Invalid Domain

Invalid email address

If all conditions satisfied then print:

Valid email address

Refer to the sample input and output for format specifications.

Sample Test Case

Input: sample@gmail.com
Output: Valid email address

Answer

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

// Custom exception for invalid dot usage
class DotException extends Exception {
    public DotException(String message) {
        super(message);
    }
}

// Custom exception for invalid @ usage
class AtTheRateException extends Exception {
    public AtTheRateException(String message) {
        super(message);
    }
}

// Custom exception for invalid domain
class DomainException extends Exception {
    public DomainException(String message) {
        super(message);
    }
}

// Main class
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String email = sc.nextLine();
        sc.close();

        try {
            validateEmail(email);
            System.out.println("Valid email address");
        } catch (DotException e) {
            System.out.println("DotException: " + e.getMessage());
            System.out.println("Invalid email address");
        } catch (AtTheRateException e) {
            System.out.println("AtTheRateException: " + e.getMessage());
            System.out.println("Invalid email address");
        } catch (DomainException e) {
            System.out.println("DomainException: " + e.getMessage());
            System.out.println("Invalid email address");
        }
    }

    public static void validateEmail(String email)
            throws DotException, AtTheRateException, DomainException {

        // Check '@' usage
        int atCount = email.length() - email.replace("@", "").length();
        if (atCount != 1 || email.startsWith("@") || email.endsWith("@") ||
email.contains("@@")) {
            throw new AtTheRateException("Invalid @ usage");
        }

        // Check '.' usage
        if (email.startsWith(".") || email.endsWith(".") || email.contains("..")) {
            throw new DotException("Invalid Dot usage");
        }

        int atIndex = email.indexOf('@');
        int dotIndex = email.lastIndexOf('.');

        if (dotIndex < atIndex || dotIndex == -1) {
            throw new DotException("Invalid Dot usage");
        }

        // Extract domain extension and validate
        String domain = email.substring(dotIndex + 1);
        if (!(domain.equals("com") || domain.equals("in") || domain.equals("net") ||
domain.equals("biz"))) {
            throw new DomainException("Invalid Domain");
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Elsa, a busy professional, is using a scheduling application to plan her meetings efficiently. The application requires users to input meeting durations in minutes, ensuring that the duration is a positive integer and does not exceed 240 minutes (4 hours). Elsa needs a program to assist her in scheduling meetings securely with proper exception handling.

Create a Java class named ElsaMeetingScheduler.Implement a custom exception:InvalidDurationException for invalid meeting duration entries.Implement the main method to interactively take user input for a meeting duration.Implement the validateMeetingDuration method to validate the meeting duration based on the specified rules and throw a custom exception if the validation fails.Print appropriate success or error messages based on the meeting duration.

Implement a custom exception, InvalidDurationException, to handle cases where the entered meeting duration does not meet the specified criteria.

Input Format

The input consists of an integer value 'n', representing the meeting duration.

Output Format

The output is displayed in the following format:

If the entered meeting duration meets the specified criteria, the program outputs

"Meeting scheduled successfully!"

If the entered meeting duration is invalid, the program outputs an error message indicating the issue.

"Error: Invalid meeting duration. Please enter a positive integer not exceeding 240 minutes (4 hours)."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 120
Output: Meeting scheduled successfully!

Answer

```
import java.util.Scanner;

// Custom exception for invalid meeting duration
class InvalidDurationException extends Exception {
    public InvalidDurationException(String message) {
        super(message);
    }
}

// Main class
 class main {
    // Method to validate meeting duration
    public static void validateMeetingDuration(int duration) throws
InvalidDurationException {
        if (duration <= 0 || duration > 240) {
            throw new InvalidDurationException(
                "Error: Invalid meeting duration. Please enter a positive integer not
exceeding 240 minutes (4 hours)."
            );
        }
    }

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

        try {
            validateMeetingDuration(duration);
            System.out.println("Meeting scheduled successfully!");
        } catch (InvalidDurationException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a user registration system, there is a requirement to implement a username validation module. Users attempting to register must adhere to specific criteria for their usernames to be considered valid.

Your task is to develop a program that takes user input for a desired username and validates it according to the following rules:

The username must not contain any spaces.The username must be at least 5 characters long.

Implement a custom exception, InvalidUsernameException, to handle cases where the entered username does not meet the specified criteria.

Input Format

The input consists of a string S, representing the desired username.

Output Format

If the username is valid, print "Username is valid: [S]".

If the username is invalid:

1. If the username is short, print "Invalid Username: Username must be at least 5 characters long"
2. If the username contains spaces, print "Invalid Username: Username cannot contain spaces"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: John

Output: Invalid Username: Username must be at least 5 characters long

Answer

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

// Custom exception for invalid usernames
class InvalidUsernameException extends Exception {
   public InvalidUsernameException(String message) {
      super(message);
   }
}

// Main class
public class Main {

   // Method to validate the username
   public static void validateUsername(String username) throws
InvalidUsernameException {
      if (username.contains(" ")) {
         throw new InvalidUsernameException("Invalid Username: Username
cannot contain spaces");
      }
```

```
        if (username.length() < 5) {
            throw new InvalidUsernameException("Invalid Username: Username must
be at least 5 characters long");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String username = sc.nextLine();
        sc.close();

        try {
            validateUsername(username);
            System.out.println("Username is valid: " + username);
        } catch (InvalidUsernameException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A local municipality is implementing an online voting system for a community event and wants to ensure that only eligible voters (those aged 18 or older) can participate.

Your task is to develop a program that validates the age of individuals attempting to vote online. If the user's age is below 18, the program should throw a custom exception, InvalidAgeException, preventing them from casting their vote. If the input is invalid, catch the appropriate InputMismatchException and print the in-built exception message.

Input Format

The input consists of an integer representing the age.

Output Format

If the age is 18 or older, print "Eligible to vote"

If the age is below 18, print "Exception occurred: InvalidAgeException: Age is not valid to vote"

If there is any other type of exception, print "An error occurred: " followed by the in-built exception message.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20
Output: Eligible to vote

Answer

```
// You are using Java
import java.util.Scanner;
import java.util.InputMismatchException;

class InvalidAgeException extends Exception {
    public InvalidAgeException(String message) {
        super(message);
    }
}

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

        try {
            int age = sc.nextInt();

            if (age < 18) {
                throw new InvalidAgeException("Age is not valid to vote");
            }

            System.out.println("Eligible to vote");
        } catch (InvalidAgeException e) {
            System.out.println("Exception occurred: InvalidAgeException: " +
e.getMessage());
```

```
    } catch (InputMismatchException e) {
      System.out.println("An error occurred: " + e);
    } catch (Exception e) {
      System.out.println("An error occurred: " + e.getMessage());
    } finally {
      sc.close();
    }
  }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
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

```
// You are using Java
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> list = new ArrayList<>();

        for (int i = 0; i < N; i++) {
            int num = sc.nextInt();


            if (list.isEmpty() || num > list.get(list.size() - 1)) {
                list.add(num);
            }
        }

        System.out.println(list);

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
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.*;

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

        LinkedList<String> playlist = new LinkedList<>();
        int currentIndex = -1;     // current song index (-1 means empty)

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

            switch (command) {
                case "ADD":
                    String songToAdd = parts[1];
                    playlist.add(songToAdd);
                    if (currentIndex == -1) {
                        currentIndex = 0;
                    }
                    break;

                case "REMOVE":
                    String songToRemove = parts[1];
                    int indexToRemove = playlist.indexOf(songToRemove);
                    if (indexToRemove != -1) {
                        playlist.remove(indexToRemove);
                        if (playlist.isEmpty()) {
                            currentIndex = -1; // empty now
                        } else if (indexToRemove < currentIndex) {
                            currentIndex--; // adjust position
                        } else if (indexToRemove == currentIndex) {
                            if (currentIndex >= playlist.size()) {
                                currentIndex = 0; // wrap around
                            }
                        }
                    }
                    break;
                case "SHOW":
                    if (playlist.isEmpty()) {
                        System.out.print("EMPTY ");
                    } else {
                        for (String s : playlist) {
                            System.out.print(s + " ");
                        }
                    }
                    break;

                case "NEXT":
                    if (playlist.isEmpty()) {
                        System.out.print("EMPTY ");
                    } else {
                        currentIndex = (currentIndex + 1) % playlist.size();
                        System.out.print(playlist.get(currentIndex) + " ");
                    }
                    break;
            }
        }
        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
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 sc = new Scanner(System.in);


    int N = sc.nextInt();
    sc.nextLine();


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


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

```
        String searchName = sc.next();


        int count = 0;
        for (String name : names) {
            if (name.equals(searchName)) {
                count++;
            }
        }


        System.out.println(count);

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
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 : 10

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 main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    int n = scanner.nextInt();
    scanner.nextLine();
```

```
        Set<String> uniqueEntries = new LinkedHashSet<>();

        for (int i = 0; i < n; i++) {
            uniqueEntries.add(scanner.nextLine());
        }


        List<String> result = new ArrayList<>(uniqueEntries);


        Collections.sort(result, new Comparator<String>() {
            @Override
            public int compare(String s1, String s2) {
                int priority1 = getPriority(s1);
                int priority2 = getPriority(s2);
                return Integer.compare(priority1, priority2);
            }

            private int getPriority(String s) {
                if (s.contains(" Bike")) return 1;
                if (s.contains(" Truck")) return 2;
                if (s.contains(" Car")) return 3;
                return 4; // default
            }
        });

        for (String entry : result) {
            System.out.println(entry);
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
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 : 10

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.Scanner;
import java.util.Map;
import java.util.HashMap;
class ValueProcessor {
    public static Map<String, Double> readValues(Scanner scanner) {
        Map<String, Double> valueMap = new HashMap<>();
        while (true) {
            String input = scanner.nextLine();
            if (input.toLowerCase().equals("done")) {
                break;
            }
            String[] pair = input.split(":");
            if (pair.length == 2) {
                String key = pair[0].trim();
                try {
                    double value = Double.parseDouble(pair[1].trim());
                    valueMap.put(key, value);
                } catch (NumberFormatException e) {
                    System.out.println("Invalid input");
                    return null;
                }
            } else {
                System.out.println("Invalid format");
                return null;
            }
        }
        return valueMap;
    }

    public static double calculateSum(Map<String, Double> valueMap) {
        double sum = 0;
        for (double value : valueMap.values()) {
            sum += value;
        }
        return sum;
    }
}
class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Map<String, Double> valueMap = ValueProcessor.readValues(scanner);
        if (valueMap != null) {
            double sum = ValueProcessor.calculateSum(valueMap);
            System.out.printf("%.2f\n", sum);
        }
        scanner.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
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 scanner = new Scanner(System.in);

    int n = Integer.parseInt(scanner.nextLine());
    TreeMap<Character, Integer> frequencyMap = new TreeMap<>();

    for (int i = 0; i < n; i++) {
      String line = scanner.nextLine();
```

```
            for (char ch : line.toCharArray()) {
                if (Character.isLetter(ch)) {
                    frequencyMap.put(ch, frequencyMap.getOrDefault(ch, 0) + 1);
                }
            }
        }

        System.out.println("Character Frequency:");
        for (Map.Entry<Character, Integer> entry : frequencyMap.entrySet()) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }

        scanner.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 11

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Project

1. Problem Statement

In ABC Corporation, employee records are stored in a database.

To efficiently manage employee details using Java and JDBC, you are tasked with building an Employee Management System that supports the following functionalities:

Adding a new employee

Updating an employee's salary

Viewing an employee's details

Displaying all employees

You are given two files:

File 1: Employee.java (POJO Class)

This class represents the Employee entity.

An Employee contains the following details:

Field	Description
employeeId	Unique Employee ID (Integer)
name	Employee Name (String)
department	Employee Department (String)
salary	Employee Salary (Double)

Students must write code in the marked area:

```
class Employee {
   private int employeeId;
   private String name;
   private String department;
   private double salary;

   public Employee() {}

   public Employee(int employeeId, String name, String department, double
salary) {
      // write your code here
   }

   // Include getters and setters
}
```

Expected in this part:

Assign parameter values to instance variables inside the constructor.

Add getters and setters for all attributes.

File 2: EmployeeDAO.java (Data Access Layer)

This class handles all database operations using JDBC.

Students must complete the missing JDBC logic in the following methods:

```
class EmployeeDAO {

   public void addEmployee(Connection conn, Employee employee) throws
SQLException {

      // write your code here

   }

   public void updateSalary(Connection conn, int employeeId, double
newSalary) throws SQLException {

      // write your code here

   }

   public void deleteEmployee(Connection conn, int employeeId) throws
SQLException {

      // write your code here

   }

   public Employee viewEmployeeRecord(Connection conn, int employeeId)
throws SQLException {

      // write your code here

   }

   public List<Employee> displayAllEmployees(Connection conn) throws
SQLException {

      // write your code here

   }
```

```
    private Employee mapToEmployee(ResultSet rs) throws SQLException {
        return new Employee(
            // write your code here
        );
    }
}
```

Expected in this part:

Write SQL queries for INSERT, UPDATE, DELETE, SELECT.

Execute queries using PreparedStatement or Statement.

Map ResultSet rows to Employee objects using mapToEmployee().

Return a List<Employee> where required.

The system should connect to a MySQL database using the following default credentials:

DB URL: jdbc:mysql://localhost/ri_dbUsername: testPassword: test123

The employees table has already been created with the following structure:

Input Format

The first line of input consists of an integer choice, representing the operation to be performed:

(1 for Add Employee, 2 for Update Salary, 3 for View Employee Record, 4 for Display All Employees, 5 for Exit)

For choice 1 (Add Employee):

1. The second line consists of an integer employee_id.
2. The third line consists of a string name.
3. The fourth line consists of a string department.
4. The fifth line consists of a double salary (must be at least 30000).

For choice 2 (Update Salary):

1. The second line consists of an integer employee_id.
2. The third line consists of a double new_salary (must be at least 30000).

For choice 3 (View Employee Record):

1. The second line consists of an integer employee_id.

For choice 4 (Display All Employees).

For choice 5 (Exit).

Output Format

For choice 1 (Add Employee),

1. Print "Employee added successfully" if the employee was added.

For choice 2 (Update Salary),

1. Print "Salary updated successfully" if the salary update was successful.
2. Print "Employee not found." if the specified employee ID does not exist.
3. Print "Salary must be at least 30000." if the provided salary is below the minimum.

For choice 3 (View Employee Record),

1. Display the employee details in the format:
2. ID: [employee_id] | Name: [name] | Department: [department] | Salary: [salary]
3. Print "Employee not found." if the specified employee ID does not exist.

For choice 4 (Display All Employees),

1. Display each employee on a new line in the format:
2. ID | Name | Department | Salary

For choice 5 (Exit),

1. Print "Exiting Employee Management System."

For invalid input:

1. Print "Invalid choice. Please try again."

Sample Test Case

```
Input: 1
101
Alice Johnson
Engineering
31000.75
4
6
5
Output: Employee added successfully
ID | Name | Department | Salary
101 | Alice Johnson | Engineering | 31000.75
Invalid choice. Please try again.
Exiting Employee Management System.
```

Answer

```
import java.sql.*;
import java.util.Scanner;

class Employee {
   private int employeeId;
   private String name;
   private String department;
   private double salary;
```

```
    // Constructor
    public Employee(int employeeId, String name, String department, double
salary) {
        this.employeeId = employeeId;
        this.name = name;
        this.department = department;
        this.salary = salary;
    }

    // Getters and Setters
    public int getEmployeeId() { return employeeId; }
    public void setEmployeeId(int employeeId) { this.employeeId = employeeId; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public String getDepartment() { return department; }
    public void setDepartment(String department) { this.department =
department; }

    public double getSalary() { return salary; }
    public void setSalary(double salary) { this.salary = salary; }
}


class EmployeeManagementSystem {

    // Add Employee
    public static void addEmployee(Connection conn, Scanner scanner) {
        int employeeId = scanner.nextInt();
        scanner.nextLine();  // Consume newline
        String name = scanner.nextLine();
        String department = scanner.nextLine();
        double salary = scanner.nextDouble();

        if (salary < 30000) {
            System.out.println("Salary must be at least 30000.");
            return;
        }

        // Create an Employee POJO object
        Employee employee = new Employee(employeeId, name, department,
salary);
```

```
        String insertQuery = "INSERT INTO employees (employee_id, name,
department, salary) VALUES (?, ?, ?, ?)";
        try (PreparedStatement stmt = conn.prepareStatement(insertQuery)) {
            stmt.setInt(1, employee.getEmployeeId());
            stmt.setString(2, employee.getName());
            stmt.setString(3, employee.getDepartment());
            stmt.setDouble(4, employee.getSalary());

            int rowsInserted = stmt.executeUpdate();
            System.out.println(rowsInserted > 0 ? "Employee added successfully" :
"Failed to add employee.");
        } catch (SQLException e) {
            System.out.println("Error adding employee: " + e.getMessage());
        }
    }

    // Update Salary
    public static void updateSalary(Connection conn, Scanner scanner) {
        int employeeId = scanner.nextInt();
        double newSalary = scanner.nextDouble();

        if (newSalary < 30000) {
            System.out.println("Salary must be at least 30000.");
            return;
        }

        String updateQuery = "UPDATE employees SET salary = ? WHERE
employee_id = ?";
        try (PreparedStatement stmt = conn.prepareStatement(updateQuery)) {
            stmt.setDouble(1, newSalary);
            stmt.setInt(2, employeeId);

            int rowsUpdated = stmt.executeUpdate();
            System.out.println(rowsUpdated > 0 ? "Salary updated successfully" :
"Employee not found.");
        } catch (SQLException e) {
            System.out.println("Error updating salary: " + e.getMessage());
        }
    }

    // View Employee Record
```

```
public static void viewEmployeeRecord(Connection conn, Scanner scanner) {
    int employeeId = scanner.nextInt();
    String selectQuery = "SELECT * FROM employees WHERE employee_id = ?";

    try (PreparedStatement stmt = conn.prepareStatement(selectQuery)) {
        stmt.setInt(1, employeeId);
        ResultSet rs = stmt.executeQuery();

        if (rs.next()) {
            Employee employee = new Employee(
                    rs.getInt("employee_id"),
                    rs.getString("name"),
                    rs.getString("department"),
                    rs.getDouble("salary")
            );
            System.out.printf("ID: %d | Name: %s | Department: %s | Salary: %.2f%n",
                    employee.getEmployeeId(),
                    employee.getName(),
                    employee.getDepartment(),
                    employee.getSalary());
        } else {
            System.out.println("Employee not found.");
        }
    } catch (SQLException e) {
        System.out.println("Error retrieving employee record: " + e.getMessage());
    }
}

// Display All Employees
public static void displayAllEmployees(Connection conn) {
    String displayQuery = "SELECT * FROM employees";

    try (Statement stmt = conn.createStatement();
         ResultSet rs = stmt.executeQuery(displayQuery)) {

        System.out.println("ID | Name | Department | Salary");
        while (rs.next()) {
            Employee employee = new Employee(
                    rs.getInt("employee_id"),
                    rs.getString("name"),
                    rs.getString("department"),
                    rs.getDouble("salary")
```

```
                );
                System.out.printf("%d | %s | %s | %.2f%n",
                        employee.getEmployeeId(),
                        employee.getName(),
                        employee.getDepartment(),
                        employee.getSalary());
            }
        } catch (SQLException e) {
            System.out.println("Error displaying employees: " + e.getMessage());
        }
    }

    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost/ri_db";
        String username = "test";
        String password = "test123";

        try (Connection conn = DriverManager.getConnection(url, username,
password);
             Scanner scanner = new Scanner(System.in)) {

            int choice;
            do {
                choice = scanner.nextInt();

                switch (choice) {
                    case 1 -> addEmployee(conn, scanner);
                    case 2 -> updateSalary(conn, scanner);
                    case 3 -> viewEmployeeRecord(conn, scanner);
                    case 4 -> displayAllEmployees(conn);
                    case 5 -> System.out.println("Exiting Employee Management
System.");
                    default -> System.out.println("Invalid choice. Please try again.");
                }

            } while (choice != 5);

        } catch (SQLException e) {
            System.out.println("Database Error: " + e.getMessage());
        }
    }
}
```

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

2. Problem Statement

Create a JDBC-based Hospital Management System that handles runtime input to manage patient records. The system should allow users to:

Add a new patient (patient ID, name, age, status).

Update a patient's status.

View a specific patient's record by patient ID.

Display all patient records in the database.

Exit the application.

The system should connect to a MySQL database using the following default credentials:

DB URL: jdbc:mysql://localhost/ri_db

USER: test

PWD: test123

The patients table has already been created with the following structure:

Table Name: patients

Input Format

The first line of input consists of an integer choice, representing the operation to be performed:

(1 for Add Patient, 2 for Update Patient Status, 3 for View Patient Record, 4 for Display All Patients, 5 for Exit)

For choice 1 (Add Patient):

- The second line consists of an integer patient_id.
- The third line consists of a string name.
- The fourth line consists of an integer age.
- The fifth line consists of a string status.

For choice 2 (Update Patient Status):

- The second line consists of an integer patient_id.
- The third line consists of a string new_status.

For choice 3 (View Patient Record):

- The second line consists of an integer patient_id.

For choice 4 (Display All Patients):

- No additional inputs are required.

For choice 5 (Exit):

- No additional inputs are required.

Output Format

For choice 1 (Add Patient):

- Print "Patient added successfully" if the patient was added.
- Print "Failed to add patient." if the insertion failed.

For choice 2 (Update Patient Status):

- Print "Patient status updated successfully" if the update was successful.
- Print "Patient not found." if the specified patient ID does not exist.

For choice 3 (View Patient Record):

- Display the patient details in the format:
- ID: [patient_id] | Name: [name] | Age: [age] | Status: [status]
- Print "Patient not found." if the specified patient ID does not exist.

For choice 4 (Display All Patients):

- Display each patient on a new line in the format:

- ID | Name | Age | Status
- If no records are available, print nothing (or handle it with an appropriate message if desired).

For choice 5 (Exit):

- Print "Exiting Hospital Management System."

For invalid input:

- Print "Invalid choice. Please try again."

Sample Test Case

```
Input: 1
101
John Doe
45
Admitted
4
5
```

```
Output: Patient added successfully
ID | Name | Age | Status
101 | John Doe | 45 | Admitted
Exiting Hospital Management System.
```

Answer

```
import java.sql.*;
import java.util.Scanner;

class HospitalManagementSystem {
   public static void main(String[] args) {
       try (Connection conn = DriverManager.getConnection("jdbc:mysql://
localhost/ri_db", "test", "test123");
            Scanner scanner = new Scanner(System.in)) {

           boolean running = true;

           while (running) {

               int choice = scanner.nextInt();

               switch (choice) {
                    case 1:
                        addPatient(conn, scanner);
                        break;
                    case 2:
                        updatePatientStatus(conn, scanner);
                        break;
                    case 3:
                        viewPatientRecord(conn, scanner);
                        break;
                    case 4:
                        displayAllPatients(conn);
                        break;
                    case 5:
                        System.out.println("Exiting Hospital Management System.");
                        running = false;
                        break;
                    default:
                        System.out.println("Invalid choice. Please try again.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }


    public static void addPatient(Connection conn, Scanner scanner) {
        int patientId = scanner.nextInt();
        scanner.nextLine();  // Consume newline

        String name = scanner.nextLine();

        int age = scanner.nextInt();

        scanner.nextLine();  // Consume newline
        String status = scanner.nextLine();

        String insertQuery = "INSERT INTO patients (patient_id, name, age, status)
VALUES (?, ?, ?, ?)";
        try (PreparedStatement stmt = conn.prepareStatement(insertQuery)) {
            stmt.setInt(1, patientId);
            stmt.setString(2, name);
            stmt.setInt(3, age);
            stmt.setString(4, status);

            int rowsInserted = stmt.executeUpdate();
            System.out.println(rowsInserted > 0 ? "Patient added successfully" :
"Failed to add patient.");
        } catch (SQLException e) {
            System.out.println("Error adding patient: " + e.getMessage());
        }
    }

    public static void updatePatientStatus(Connection conn, Scanner scanner) {
        int patientId = scanner.nextInt();
        scanner.nextLine();  // Consume newline

        String newStatus = scanner.nextLine();

        String updateQuery = "UPDATE patients SET status = ? WHERE patient_id
= ?";
        try (PreparedStatement stmt = conn.prepareStatement(updateQuery)) {
            stmt.setString(1, newStatus);
            stmt.setInt(2, patientId);

            int rowsUpdated = stmt.executeUpdate();
            System.out.println(rowsUpdated > 0 ? "Patient status updated
successfully" : "Patient not found.");
        } catch (SQLException e) {
            System.out.println("Error updating patient status: " + e.getMessage());
        }
    }

    public static void viewPatientRecord(Connection conn, Scanner scanner) {
        int patientId = scanner.nextInt();

        String selectQuery = "SELECT * FROM patients WHERE patient_id = ?";
        try (PreparedStatement stmt = conn.prepareStatement(selectQuery)) {
            stmt.setInt(1, patientId);

            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                System.out.printf("ID: %d | Name: %s | Age: %d | Status: %s%n",
                        rs.getInt("patient_id"),
                        rs.getString("name"),
                        rs.getInt("age"),
                        rs.getString("status"));
            } else {
                System.out.println("Patient not found.");
            }
        } catch (SQLException e) {
            System.out.println("Error retrieving patient record: " + e.getMessage());
        }
    }

    public static void displayAllPatients(Connection conn) {
        String displayQuery = "SELECT * FROM patients ORDER BY patient_id";
        try (Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(displayQuery)) {

            System.out.println("ID | Name | Age | Status");
            while (rs.next()) {
                System.out.printf("%d | %s | %d | %s%n",
                        rs.getInt("patient_id"),
                        rs.getString("name"),
                        rs.getInt("age"),
                        rs.getString("status"));
            }
        } catch (SQLException e) {
            System.out.println("Error displaying patients: " + e.getMessage());
        }
    }

}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_Week 12_Java_Lamba Expressions_MCQ

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : MCQ

1. Which of the following interfaces is NOT a functional interface in Java?

Answer

Iterable

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

2. Which functional interface is commonly used with lambda expressions in Java?

Answer

Runnable

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

3. Which of the following is a valid lambda expression in Java?

Answer

All of the mentioned options

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

4. What is the syntax for a basic lambda expression in Java?

Answer

(parameters) -&gt; expression

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

5. Can a lambda expression have more than one parameter?

Answer

Yes, it can have multiple parameters

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

6. Can a lambda expression in Java have a body with multiple statements?

Answer

Yes, if the statements are enclosed in curly braces

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

7. Which functional interface in Java takes two arguments and returns a result?

Answer

BiFunction

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

8. What is the return type of a lambda expression in Java?

Answer

The return type is inferred from the context

Status : Correct *Marks : 1/1*

9. Can a lambda expression in Java have a body with multiple statements?

Answer

Yes, if the statements are enclosed in curly braces

Status : Correct *Marks : 1/1*

10. What is a lambda expression in Java?

Answer

A way to define anonymous methods

Status : Correct *Marks : 1/1*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sabrina is working on a project that involves analyzing a set of numbers. In her exploration, she encounters scenarios where extracting even numbers and finding their sum is essential.

Create a program that calculates the sum of even numbers from a given array of integers using a lambda expression.

Input Format

The first line of input consists of an integer N, representing the size of the array.

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

Output Format

The output prints the sum of the even integers from the array.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
29 37 45

Output: 0

Answer

```
import java.util.Scanner;
class EvenSumCalculator {
   public static int calculateEvenSum(int[] numbers) {
      return java.util.Arrays.stream(numbers)
            .filter(n -> n % 2 == 0)
            .sum();
   }
}
class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      int count = scanner.nextInt();
      int[] numbers = new int[count];

      for (int i = 0; i < count; i++) {
         numbers[i] = scanner.nextInt();
      }
      int sum = EvenSumCalculator.calculateEvenSum(numbers);
      System.out.println(sum);

      scanner.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Alex is learning about Java's functional interfaces and lambda expressions.

He wants to write a simple program that prints the square of each number in an array using a predefined functional interface.

Help Alex complete this task using the Consumer functional interface.

Input Format

- The first line contains an integer N, the number of elements in the array.
- The second line contains N space-separated integers.

Output Format

- Print the squares of all elements in the array, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
1 2 3 4

Output: 1 4 9 16

Answer

```
import java.util.*;
import java.util.function.Consumer;

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

      // Read number of elements
      int n = sc.nextInt();
      int[] arr = new int[n];

      // Read array elements
      for (int i = 0; i < n; i++) {
         arr[i] = sc.nextInt();
      }

      // Define a Consumer functional interface using lambda expression
      Consumer<Integer> printSquare = x -> System.out.print((x * x) + " ");

      // Apply the lambda for each element
      for (int num : arr) {
         printSquare.accept(num);
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In the mystical realm of programming, there exists a magical incantation to reveal hidden words.

Elara, the skilled enchantress, wishes to summon a word using her spell and then reverse its characters to uncover its enchanted reflection.

Write a program that uses the predefined functional interface Supplier<String> and a lambda expression to:

Supply (generate) a string, and

Display its reversed form.

Input Format

No input is required from the user.

The string must be supplied internally using a Supplier<String>.

Output Format

Print the reversed version of the supplied string.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Wizard!!
Output: !!draziW

Answer

```
import java.util.Scanner;

public class Main {

   // Interface for the Lambda Expression
   interface ReverseStringFunction {
      String reverse(String input);
   }

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      String input = scanner.nextLine();

      // Use Lambda Expression to reverse the string
      String reversed = ((ReverseStringFunction) s -> new
StringBuilder(s).reverse().toString()).reverse(input);

      System.out.println(reversed);

      scanner.close();
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Abi is working on a text analysis project where she needs to categorize words based on their length.

Words that have three or fewer characters are considered “Short”, while

words with more than three characters are classified as “Long.”

Write a Java program that takes a sentence as input, analyzes each word, and prints a list showing whether each word is “Short” or “Long.”

Use the predefined functional interface Function<String, String> along with a lambda expression for categorization.

Input Format

A single line containing a sentence (words separated by spaces).

Output Format

- A single line with each word categorized as "Short" or "Long", separated by spaces.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: I love my cat
Output: Short Long Short Short

Answer

```
import java.util.*;
import java.util.function.Function;

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

        // Read input sentence
        String sentence = sc.nextLine();

        // Split sentence into words
        String[] words = sentence.split(" ");

        // Define Function interface using lambda
        Function<String, String> categorize = word ->
            (word.length() <= 3) ? "Short" : "Long";

        // Analyze and print results
        for (String word : words) {
            System.out.print(categorize.apply(word) + " ");
        }
    }
}
```

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

Character Arrays and Strings

What is the output of the following program?

```c
#include <stdio.h>
int main()
{
      char s1[10] = "REC";
      char s2[10] = "REC";
      if (s1 == s2)
            printf("Equal");
      else
            printf("Not Equal");
      return 0;
}
```

A. Equal
B. Not Equal
C. Compile time error
D. None of the mentioned

ANSWER: B

What is the output of this C code?

```
#include <stdio.h>
int main()
{
      char str[] = "hello, world";
      str[5] = '.';
      printf("%s", str);
      return 0;
}
```

A. hello. world
B. hello, world
C. Compile error
D. Segmentation fault

ANSWER: A

Find the output of the following C program.

```
#include <stdio.h>
int main()
{
	char a[10][8] = { "hi", "hello", "fellows" };
	printf("%s", a[2]);
	return 0;
}
```

A. hi
B. hello
C. fellows
D. Segmentation fault

ANSWER: C

a[2] indicates the 3rd string of the 2D array. Thus “fellows” will be printed.

What will be the output of the C code?

```
#include <stdio.h>
int main()
{
	char arr[] = "C Programming";
	printf("%d", sizeof(arr));
	return 0;
}
```

A. 12
B. 13
C. 14
D. Segmentation fault

ANSWER: C

size of arr[] is 13 as it is '\0' terminated

What will be output?

```
# include <stdio.h>
int main()
{
      char str1[] = "Hello World";
      char str2[] = { 'H','e','l','l','o',' ','W','o','r','l','d' };
      int n1 = sizeof(str1) / sizeof(str1[0]);
      int n2 = sizeof(str2) / sizeof(str2[0]);
      printf("%d,%d", n1, n2);
      return 0;
}
```

A. 12,12
B. 11,11
C. 12,11
D. Segmentation fault

ANSWER: C

What will be the output of the following programs?

```
#include <stdio.h>
int main()
{
	char c[2] = "A";
	printf("%c ", c[0]);
	printf("%s", c);
	return 0;
}
```

A. A A
B. 65 A
C. A 65
D. 65 65

ANSWER: A

What will be the output of the following programs?

```
#include <stdio.h>
int main()
{
      char s[] = "Get organised! learn C!!";
      printf("%s\n", &s[2]);
      return 0;
}
```

A. t organised! learn C!!
B. Get organised! learn C!!
C. t
D. Segmentation Fault

ANSWER: A

What will be the output of the following programs?

```
#include <stdio.h>
int main()
{
	char s[] = "Get organised! learn C!!";
	printf("%s", s);
	return 0;
}
```

A. t organised! learn C!!
B. Get organised! learn C!!
C. t
D. Segmentation Fault

ANSWER: B

What will be the output of the following programs?

```
#include <stdio.h>
int main()
{
      char s[] = "Get organised! learn C!!";
      printf("%s", &s);
      return 0;
}
```

A. t organised! learn C!!
B. Get organised! learn C!!
C. t
D. Segmentation Fault

ANSWER: B

What will be the output of the following programs?

```
#include <stdio.h>
int main()
{
	char s[] = "Get organised! learn C!!";
	printf("%c", s[2]);
	return 0;
}
```

A. t organised! learn C!!
B. Get organised! learn C!!
C. t
D. Segmentation Fault

ANSWER: C

What will be the output of the following programs?

```
#include <stdio.h>
int main()
{
     char str1[] = { 'H', 'e', 'l', 'l', 'o', 0 };
     char str2[] = "Hello";
     printf("%s ", str1);
     printf("%s", str2);
     return 0;
}
```

A. Hello Hello
B. H e l l o Hello
C. H e l l o 0 Hello
D. Garbage values Hello

ANSWER: A

What will be the output of the following programs?

```
#include <stdio.h>

int main()
{
	printf(5 + "Good Morning ");
	return 0;
}
```

A. Good
B. Morning
C. Good Morning
D. M

ANSWER: B

Explanation:
printf(5 + "Good Morning\n");
It skips the 5 characters and prints the given string.
Hence the output is "Morning".

What will be the output of the following programs?

```c
#include <stdio.h>
int main()
{
	printf("%c", "abcdefgh"[4]);
	return 0;
}
```

A. abcde
B. e
C. efgh
D. d

ANSWER: B

Explanation:
printf("%c\n", "abcdefgh"[4]);
It prints the 5 character of the string "abcdefgh".
Hence the output is 'e'.

What will be the output of the following programs?

```
#include <stdio.h>
int main()
{
      printf("%d %d %d\n", sizeof('3'), sizeof("3"), sizeof(3));
      return 0;
}
```

A. 4 2 4
B. 1 1 1
C. 2 2 2
D. 0 0 0
ANSWER: A

What will be the output of the program?

```
#include <stdio.h>
int main()
{
      char str[] = "India\0\REC\0";
      printf("%s", str);
      return 0;
}
```

A. REC
B. India
C. India REC
D. India\0REC

ANSWER: B

Explanation:
A string is a collection of characters terminated by '\0'.
Step 1:
char str[] = "India\0\REC\0";
The variable str is declared as an array of characters and initialized with value "India".
Step 2:
printf("%s", str);
It prints the value of the str.
The output of the program is "India".

What will be the output of the program?

```
#include <stdio.h>
int main()
{
    printf("India", "REC");
    return 0;
}
```

A. Error
B. India REC
C. India
D. REC

ANSWER: C

Explanation:
printf("India", "REC");
It prints "India".
Because , (comma) operator has Left to Right associativity. After printing "India", the statement got terminated.

What will be the output of the program?

```
#include <stdio.h>
int main()
{
    char str[7] = "IndiaREC";
    printf("%s", str);
    return 0;
}
```

A. Error
B. IndiaREC
C. Cannot predict
D. None of above

ANSWER: C

Explanation:
Here str[] has declared as 7 character array and into a 8 character is stored. This will result in overwriting of the byte beyond 7 byte reserved for '\0'.

What will be the output of the program?

```c
#include <stdio.h>
int main()
{
    char a[] = "\0";
    if(printf("%s", a))
        printf("The string is empty");
    else
        printf("The string is not empty");
    return 0;
}
```

A. The string is empty
B. The string is not empty
C. No output
D. 0

ANSWER: B

Explanation:
The function printf() returns the number of charecters printed on the console.
Step 1:
char a[] = "\0";
The variable a is declared as an array of characters and it initialized with "\0". It denotes that the string is empty.
Step 2:
if(printf("%s", a))
The printf() statement does not print anything, so it returns '0'(zero). Hence the if condition is failed. In the else part it prints "The string is not empty".

What will be the output of the program?

```
#include <stdio.h>
int main()
{
    char str[] = "Nagpur";
    str[0] = 'K';
    printf("%s, ", str);
    str = "Kanpur";
    printf("%s", str + 1);
    return 0;
}
```

A. Kagpur, Kanpur
B. Nagpur, Kanpur
C. Kagpur, anpur
D. Error

ANSWER: D

Explanation:
The statement
str = "Kanpur";
generates the LVALUE required error. We have to use strcpy function to copy a string.
To remove error we have to change this statement
str = "Kanpur"; tostrcpy(str, "Kanpur");
The program prints the string "anpur"

What will be the output of the C code?

```
#include <stdio.h>
int main()
{
	char str[20];
	int i;
	for (i = 0; i <= 18; i++)
		i[str] = 'C';
	i[str] = '\0';
	printf("%s", str);
	return 0;
}
```

A. CCCCCCCCCCCCCCCCCCC
B. C
C. 0
D. Error

ANSWER: A

What will be the output of the C program?

```
#include <stdio.h>
int main()
{
	int i = 0;
	printf("Hello");
	char s[4] = { '\b', '\t', '\r', '\n' };
	for (i = 0; i < 4; i++)
	{
		printf("%c", s[i]);
	}
	return 0;
}
```

A. Hello
B. Compilation error
C. Hell
D. None of the above
ANSWER: C

Explanation
Hello is printed followed by \b\t\r\n.
i.e) Hello\b\t\r\n.
i.e) Hell\t\r\n.
i.e) Hell   \r\n.
i.e) Hell\n.
i.e) Hell is Outputted.

What will be the output of the C program?

```
#include <stdio.h>
int main()
{
	int i = 0;
	char s[4] = { '\0', '\0', '\0', '\0' };
	for (i = 0; i < 4; i++)
	{
		printf("%c", s[i]);
	}
	return 0;
}
```

A. \0 \0 \0
B. \0 \0 \0 \0
C. No output
D. None of the above
ANSWER: C

Explanation
\0 = NULL. Thus compiler prints nothing.

What will be the output of the C program?

```c
#include <stdio.h>
int main()
{
     int i = 0;
     printf("Hello");
     char s[4] = { '\b', '\r', '\t', '\n' };
     for (i = 0; i < 4; i++)
     {
          printf("%c", s[i]);
     }
     return 0;
}
```

A. Hello
B. Hell
C. No output
D. Compilation error
ANSWER: C

Explanation
Hello is printed followed by \b\r\t\n.
i.e) Hello\b\r\t\n.
i.e) Hell\r\t\n.
i.e) \t\n.
i.e)   \n.
i.e)   is Outputted. ie (8 space is outputted)

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_MCQ_Updated

Attempt : 1
Total Mark : 20
Marks Obtained : 18

Section 1 : MCQ

1. Which of the following statements correctly creates a new node for a doubly linked list?

Answer

struct Node newNode = (struct Node) malloc(sizeof(struct Node));

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

2. How do you reverse a doubly linked list?

Answer

By swapping the next and previous pointers of each node

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

3. Consider the following function that refers to the head of a Doubly Linked List as the parameter. Assume that a node of a doubly linked list has the previous pointer as prev and the next pointer as next.

Assume that the reference of the head of the following doubly linked list is passed to the below function 1 <--> 2 <--> 3 <--> 4 <--> 5 <-->6. What should be the modified linked list after the function call?

```
Procedure fun(head_ref: Pointer to Pointer of node)
   temp = NULL
   current = *head_ref

   While current is not NULL
      temp = current->prev
      current->prev = current->next
      current->next = temp
      current = current->prev
   End While

   If temp is not NULL
      *head_ref = temp->prev
   End If
End Procedure
```

Answer

6 &lt;--&gt; 5 &lt;--&gt; 4 &lt;--&gt; 3 &lt;--&gt; 2 &lt;--&gt; 1.

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

4. What will be the effect of setting the prev pointer of a node to NULL in a doubly linked list?

Answer

The node will become the new head

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

5. What is the main advantage of a two-way linked list over a one-way

linked list?

Answer

Two-way linked lists allow for traversal in both directions.

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

6. Which of the following is false about a doubly linked list?

Answer

Implementing a doubly linked list is easier than singly linked list

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

7. What happens if we insert a node at the beginning of a doubly linked list?

Answer

The previous pointer of the new node is NULL

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

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

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
   struct Node* prev;
};

int main() {
   struct Node* head = NULL;
   struct Node* tail = NULL;
   for (int i = 0; i < 5; i++) {
        struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
        temp->data = i + 1;
        temp->prev = tail;
        temp->next = NULL;
        if (tail != NULL) {
            tail->next = temp;
        } else {
            head = temp;
        }
        tail = temp;
    }
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    return 0;
}
```

Answer

1 2 3 4 5

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

9. What is a memory-efficient double-linked list?

Answer

A doubly linked list that uses bitwise AND operator for storing addresses

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

10. Which of the following information is stored in a doubly-linked list's nodes?

Answer

All of the mentioned options

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

11. How do you delete a node from the middle of a doubly linked list?

Answer

All of the mentioned options

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

12. Which pointer helps in traversing a doubly linked list in reverse order?

Answer

prev

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

13. What does the following code snippet do?

```
struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
newNode->data = value;
newNode->next = NULL;
newNode->prev = NULL;
```

Answer

Creates a new node and initializes its data to 'value'

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

14. Which of the following is true about the last node in a doubly linked list?

Answer

Its next pointer is NULL

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

15. Consider the provided pseudo code. How can you initialize an empty two-way linked list?

```
Define Structure Node
   data: Integer
   prev: Pointer to Node
   next: Pointer to Node
End Define

Define Structure TwoWayLinkedList
   head: Pointer to Node
   tail: Pointer to Node
End Define
```

Answer

struct TwoWayLinkedList* list = malloc(sizeof(struct TwoWayLinkedList)); list-&gt;head = NULL; list-&gt;tail = NULL;

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

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

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
   struct Node* prev;
};

int main() {
   struct Node* head = NULL;
   struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
   temp->data = 2;
   temp->next = NULL;
   temp->prev = NULL;
   head = temp;
   printf("%d\n", head->data);
   free(temp);
   return 0;
}
```

Answer

2

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

17. What is the correct way to add a node at the beginning of a doubly linked list?

Answer

void addFirst(int data){   Node* newNode = new Node(data);   newNode-&gt;next = head;   if (head != NULL) {   head-&gt;prev = newNode;   }   head = newNode;   }

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

18. How many pointers does a node in a doubly linked list have?

Answer

2

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

19. Which code snippet correctly deletes a node with a given value from a doubly linked list?

```
void deleteNode(Node** head_ref, Node* del_node) {
   if (*head_ref == NULL || del_node == NULL) {
      return;
   }
   if (*head_ref == del_node) {
      *head_ref = del_node->next;
   }
   if (del_node->next != NULL) {
      del_node->next->prev = del_node->prev;
   }
   if (del_node->prev != NULL) {
      del_node->prev->next = del_node->next;
```

```
    }
    free(del_node);
}
```

Answer

Deletes the first occurrence of a given data value in a doubly linked list.

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

20. Where Fwd and Bwd represent forward and backward links to the adjacent elements of the list. Which of the following segments of code deletes the node pointed to by X from the doubly linked list, if it is assumed that X points to neither the first nor the last node of the list?

A doubly linked list is declared as

```
struct Node {
      int Value;
      struct Node *Fwd;
      struct Node *Bwd;
);
```

Answer

X-&gt;Bwd.Fwd = X-&gt;Fwd ; X.Fwd-&gt;Bwd = X-&gt;Bwd;

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 11

Section 1 : MCQ

1. Which of the following is the correct post-order traversal of a binary search tree with nodes: 50, 30, 20, 55, 32, 52, 57?

Answer

20, 32, 30, 52, 57, 55, 50

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

2. Find the preorder traversal of the given binary search tree.

Answer

9, 2, 1, 6, 4, 7, 10, 14

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

3. While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is _______.

Answer

67

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

4. Find the in-order traversal of the given binary search tree.

Answer

13, 2, 1, 4, 14, 18

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

5. In a binary search tree with nodes 18, 28, 12, 11, 16, 14, 17, what is the value of the left child of the node 16?

Answer

14

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

6. The preorder traversal of a binary search tree is 15, 10, 12, 11, 20, 18, 16, 19. Which one of the following is the postorder traversal of the tree?

Answer

11, 12, 10, 16, 19, 18, 20, 15

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

7. While inserting the elements 5, 4, 2, 8, 7, 10, 12 in a binary search tree, the element at the lowest level is ________.

Answer

12

Status : Correct *Marks : 1/1*

8. Find the post-order traversal of the given binary search tree.

Answer

17, 10, 18, 20, 15, 32, 21

Status : Wrong *Marks : 0/1*

9. How many distinct binary search trees can be created out of 4 distinct keys?

Answer

14

Status : Correct *Marks : 1/1*

10. Which of the following is the correct in-order traversal of a binary search tree with nodes: 9, 3, 5, 11, 8, 4, 2?

Answer

2, 3, 4, 5, 8, 9, 11

Status : Correct *Marks : 1/1*

11. Which of the following operations can be used to traverse a Binary Search Tree (BST) in ascending order?

Answer

Inorder traversal

Status : Correct *Marks : 1/1*

12. Find the pre-order traversal of the given binary search tree.

Answer

13, 2, 1, 4, 14, 18

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

13. Which of the following is the correct pre-order traversal of a binary search tree with nodes: 50, 30, 20, 55, 32, 52, 57?

Answer

50, 30, 20, 55, 32, 57, 52

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

14. Find the postorder traversal of the given binary search tree.

Answer

1, 2, 4, 13, 14, 18

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

15. Which of the following is a valid preorder traversal of the binary search tree with nodes: 18, 28, 12, 11, 16, 14, 17?

Answer

18, 12, 11, 16, 14, 17, 28

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_MCQ_Updated

Attempt : 1
Total Mark : 20
Marks Obtained : 18

Section 1 : MCQ

1. Which of the following statements correctly creates a new node for a doubly linked list?

Answer

struct Node newNode = (struct Node) malloc(sizeof(struct Node));

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

2. How do you reverse a doubly linked list?

Answer

By swapping the next and previous pointers of each node

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

3. Consider the following function that refers to the head of a Doubly Linked List as the parameter. Assume that a node of a doubly linked list has the previous pointer as prev and the next pointer as next.

Assume that the reference of the head of the following doubly linked list is passed to the below function 1 <--> 2 <--> 3 <--> 4 <--> 5 <-->6. What should be the modified linked list after the function call?

```
Procedure fun(head_ref: Pointer to Pointer of node)
   temp = NULL
   current = *head_ref

   While current is not NULL
      temp = current->prev
      current->prev = current->next
      current->next = temp
      current = current->prev
   End While

   If temp is not NULL
      *head_ref = temp->prev
   End If
End Procedure
```

Answer

6 &lt;--&gt; 5 &lt;--&gt; 4 &lt;--&gt; 3 &lt;--&gt; 2 &lt;--&gt; 1.

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

4. What will be the effect of setting the prev pointer of a node to NULL in a doubly linked list?

Answer

The node will become the new head

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

5. What is the main advantage of a two-way linked list over a one-way

linked list?

Answer

Two-way linked lists allow for traversal in both directions.

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

6. Which of the following is false about a doubly linked list?

Answer

Implementing a doubly linked list is easier than singly linked list

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

7. What happens if we insert a node at the beginning of a doubly linked list?

Answer

The previous pointer of the new node is NULL

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

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

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
   struct Node* prev;
};

int main() {
   struct Node* head = NULL;
   struct Node* tail = NULL;
   for (int i = 0; i < 5; i++) {
        struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
        temp->data = i + 1;
        temp->prev = tail;
        temp->next = NULL;
        if (tail != NULL) {
            tail->next = temp;
        } else {
            head = temp;
        }
        tail = temp;
    }
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    return 0;
}
```

Answer

1 2 3 4 5

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

9. What is a memory-efficient double-linked list?

Answer

A doubly linked list that uses bitwise AND operator for storing addresses

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

10. Which of the following information is stored in a doubly-linked list's nodes?

Answer

All of the mentioned options

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

11. How do you delete a node from the middle of a doubly linked list?

Answer

All of the mentioned options

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

12. Which pointer helps in traversing a doubly linked list in reverse order?

Answer

prev

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

13. What does the following code snippet do?

```
struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
newNode->data = value;
newNode->next = NULL;
newNode->prev = NULL;
```

Answer

Creates a new node and initializes its data to 'value'

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

14. Which of the following is true about the last node in a doubly linked list?

Answer

Its next pointer is NULL

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

15. Consider the provided pseudo code. How can you initialize an empty two-way linked list?

```
Define Structure Node
   data: Integer
   prev: Pointer to Node
   next: Pointer to Node
End Define

Define Structure TwoWayLinkedList
   head: Pointer to Node
   tail: Pointer to Node
End Define
```

Answer

struct TwoWayLinkedList* list = malloc(sizeof(struct TwoWayLinkedList)); list-&gt;head = NULL; list-&gt;tail = NULL;

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

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

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
   struct Node* prev;
};

int main() {
   struct Node* head = NULL;
   struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
   temp->data = 2;
   temp->next = NULL;
   temp->prev = NULL;
   head = temp;
   printf("%d\n", head->data);
   free(temp);
   return 0;
}
```

Answer

2

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

17. What is the correct way to add a node at the beginning of a doubly linked list?

Answer

```
void addFirst(int data){    Node* newNode = new Node(data);     newNode-
&gt;next = head;            if (head != NULL) {                  head-&gt;prev =
newNode;    }    head = newNode;           }
```

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

18. How many pointers does a node in a doubly linked list have?

Answer

2

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

19. Which code snippet correctly deletes a node with a given value from a doubly linked list?

```
void deleteNode(Node** head_ref, Node* del_node) {
   if (*head_ref == NULL || del_node == NULL) {
      return;
   }
   if (*head_ref == del_node) {
      *head_ref = del_node->next;
   }
   if (del_node->next != NULL) {
      del_node->next->prev = del_node->prev;
   }
   if (del_node->prev != NULL) {
      del_node->prev->next = del_node->next;
```

```
    }
    free(del_node);
}
```

Answer

Deletes the first occurrence of a given data value in a doubly linked list.

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

20. Where Fwd and Bwd represent forward and backward links to the adjacent elements of the list. Which of the following segments of code deletes the node pointed to by X from the doubly linked list, if it is assumed that X points to neither the first nor the last node of the list?

A doubly linked list is declared as

```
struct Node {
      int Value;
      struct Node *Fwd;
      struct Node *Bwd;
);
```

Answer

X-&gt;Bwd.Fwd = X-&gt;Fwd ; X.Fwd-&gt;Bwd = X-&gt;Bwd;

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_MCQ_Updated

Attempt : 1
Total Mark : 20
Marks Obtained : 17

Section 1 : MCQ

1. After performing this set of operations, what does the final list look to contain?

```
InsertFront(10);
InsertFront(20);
InsertRear(30);
DeleteFront();
InsertRear(40);
InsertRear(10);
DeleteRear();
InsertRear(15);
display();
```

Answer

10 30 40 15

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

2. What does the front pointer in a linked list implementation of a queue contain?

Answer

The address of the first element

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

3. What will the output of the following code?

```
#include <stdio.h>
#include <stdlib.h>
typedef struct {
   int* arr;
   int front;
   int rear;
   int size;
} Queue;
Queue* createQueue() {
   Queue* queue = (Queue*)malloc(sizeof(Queue));
   queue->arr = (int*)malloc(5 * sizeof(int));
   queue->front = 0;
   queue->rear = -1;
   queue->size = 0;
   return queue;
}
int main() {
   Queue* queue = createQueue();
   printf("%d", queue->size);
   return 0;
}
```

Answer

0

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

4. Which one of the following is an application of Queue Data Structure?

Answer

When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes

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

5. The essential condition that is checked before insertion in a queue is?

Answer

Overflow

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

6. What are the applications of dequeue?

Answer

All the mentioned options

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

7. A normal queue, if implemented using an array of size MAX_SIZE, gets full when

Answer

Rear = MAX_SIZE – 1

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

8. In what order will they be removed If the elements “A”, “B”, “C” and “D” are placed in a queue and are deleted one at a time

Answer

ABCD

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

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

```
#include <stdio.h>
#define MAX_SIZE 5
typedef struct {
    int arr[MAX_SIZE];
    int front;
    int rear;
    int size;
} Queue;

void enqueue(Queue* queue, int data) {
    if (queue->size == MAX_SIZE) {
        return;
    }
    queue->rear = (queue->rear + 1) % MAX_SIZE;
    queue->arr[queue->rear] = data;
    queue->size++;
}
int dequeue(Queue* queue) {
    if (queue->size == 0) {
        return -1;
    }
    int data = queue->arr[queue->front];
    queue->front = (queue->front + 1) % MAX_SIZE;
    queue->size--;
    return data;
}
int main() {
    Queue queue;
    queue.front = 0;
    queue.rear = -1;
    queue.size = 0;
    enqueue(&queue, 1);
    enqueue(&queue, 2);
    enqueue(&queue, 3);
    printf("%d ", dequeue(&queue));
    printf("%d ", dequeue(&queue));
    enqueue(&queue, 4);
    enqueue(&queue, 5);
    printf("%d ", dequeue(&queue));
    printf("%d ", dequeue(&queue));
    return 0;
}
```

Answer

3 2 1 5

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

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

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_SIZE 5
typedef struct {
    int* arr;
    int front;
    int rear;
    int size;
} Queue;
Queue* createQueue() {
    Queue* queue = (Queue*)malloc(sizeof(Queue));
    queue->arr = (int*)malloc(MAX_SIZE * sizeof(int));
    queue->front = -1;
    queue->rear = -1;
    queue->size = 0;
    return queue;
}
int isEmpty(Queue* queue) {
    return (queue->size == 0);
}
int main() {
    Queue* queue = createQueue();
    printf("Is the queue empty? %d", isEmpty(queue));
    return 0;
}
```

Answer

Is the queue empty? 1

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

11. Which of the following can be used to delete an element from the front end of the queue?

Answer

```
public Object deleteFront() throws emptyDEQException{if(isEmpty())throw new
emptyDEQException("Empty");else{Node temp = head.getNext();Node cur =
temp.getNext();Object e = temp.getEle();head.setNext(temp);size--;return e;}}
```

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

12. When new data has to be inserted into a stack or queue, but there is no available space. This is known as

Answer

overflow

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

13. Which of the following properties is associated with a queue?

Answer

First In First Out

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

14. In a linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into a non-empty queue?

Answer

Only rear pointer

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

15. Front and rear pointers are tracked in the linked list implementation of a queue. Which of these pointers will change during an insertion into the EMPTY queue?

Answer

Both front and rear pointer

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

16. Which operations are performed when deleting an element from an array-based queue?

Answer

Dequeue

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

17. What is the functionality of the following piece of code?

```
public void function(Object item)
{
  Node temp=new Node(item,trail);
  if(isEmpty())
  {
    head.setNext(temp);
    temp.setNext(trail);
  }
  else
  {
    Node cur=head.getNext();
    while(cur.getNext()!=trail)
    {
      cur=cur.getNext();
    }
    cur.setNext(temp);
  }
  size++;
}
```

Answer

Insert at the rear end of the dequeue

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

18. In linked list implementation of a queue, the important condition for a queue to be empty is?

Answer

FRONT is null

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

19. Insertion and deletion operation in the queue is known as

Answer

Enqueue and Dequeue

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

20. The process of accessing data stored in a serial access memory is similar to manipulating data on a

Answer

Queue

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_MCQ_Updated_1

Attempt : 1
Total Mark : 20
Marks Obtained : 18

Section 1 : MCQ

1. Which of the following is not true about QuickSort?

Answer

It as an adaptive sorting algorithm

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

2. What is the best sorting algorithm to use for the elements in an array that are more than 1 million in general?

Answer

Quick sort.

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

3. Which of the following methods is used for sorting in merge sort?

Answer

merging

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

4. In a quick sort algorithm, what role does the pivot element play?

Answer

It is used to partition the array

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

5. Is Merge Sort a stable sorting algorithm?

Answer

Yes, always stable.

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

6. Which of the following scenarios is Merge Sort preferred over Quick Sort?

Answer

When sorting linked lists

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

7. The following code snippet is an example of a quick sort. What do the 'low' and 'high' parameters represent in this code?

```
void quickSort(int arr[], int low, int high) {
   if (low < high) {
      int pivot = partition(arr, low, high);
      quickSort(arr, low, pivot - 1);
      quickSort(arr, pivot + 1, high);
```

```
    }
}
```

Answer

The range of elements to sort within the array

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

8. In a quick sort algorithm, where are smaller elements placed to the pivot during the partition process, assuming we are sorting in increasing order?

Answer

To the left of the pivot

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

9. Let P be a quick sort program to sort numbers in ascending order using the first element as a pivot. Let t1 and t2 be the number of comparisons made by P for the inputs {1, 2, 3, 4, 5} and {4, 1, 5, 3, 2}, respectively. Which one of the following holds?

Answer

t1 &lt; t2

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

10. Which of the following modifications can help Quicksort perform better on small subarrays?

Answer

Switching to Insertion Sort for small subarrays

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

11. Why is Merge Sort preferred for sorting large datasets compared to Quick Sort?

Answer

Merge Sort has better worst-case time complexity

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

12. What is the main advantage of Quicksort over Merge Sort?

Answer

Quicksort requires less auxiliary space

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

13. Merge sort is _______.

Answer

Comparison-based sorting algorithm

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

14. Which of the following sorting algorithms is based on the divide and conquer method?

Answer

Merge Sort

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

15. Which of the following strategies is used to improve the efficiency of Quicksort in practical implementations?

Answer

Choosing the pivot randomly or using the median-of-three method

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

16. What happens during the merge step in Merge Sort?

Answer

Two sorted subarrays are combined into one sorted array

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

17. Which of the following is true about Quicksort?

Answer

It is an in-place sorting algorithm

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

18. Consider the Quick Sort algorithm, which sorts elements in ascending order using the first element as a pivot. Then which of the following input sequences will require the maximum number of comparisons when this algorithm is applied to it?

Answer

22 25 56 67 89

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

19. Which of the following statements is true about the merge sort algorithm?

Answer

It requires additional memory for merging

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

20. What happens when Merge Sort is applied to a single-element array?

Answer

The array remains unchanged and no merging is required

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_MCQ_Updated

Attempt : 1
Total Mark : 20
Marks Obtained : 0

Section 1 : MCQ

1. What happens if we do not use modular arithmetic in linear probing?

Answer

Status : Skipped *Marks : 0/1*

2. Which folding method divides the key into equal parts, reverses some of them, and then adds all parts?

Answer

-

Status : - *Marks : 0/1*

3. Which of the following statements is TRUE regarding the folding method?

Answer

-

Status : - *Marks : 0/1*

4. Which of the following best describes linear probing in hashing?

Answer

-

Status : - *Marks : 0/1*

5. What is the primary disadvantage of linear probing?

Answer

-

Status : - *Marks : 0/1*

6. Which of these hashing methods may result in more uniform distribution with small keys?

Answer

-

Status : - *Marks : 0/1*

7. In linear probing, if a collision occurs at index i, what is the next index checked?

Answer

-

Status : - *Marks : 0/1*

8. What is the worst-case time complexity for inserting an element in a hash table with linear probing?

Answer

-

Status : - *Marks : 0/1*

9. What does a deleted slot in linear probing typically contain?

Answer

-

Status : - *Marks : 0/1*

10. What is the initial position for a key k in a linear probing hash table?

Answer

-

Status : - *Marks : 0/1*

11. Which situation causes clustering in linear probing?

Answer

-

Status : - *Marks : 0/1*

12. Which C statement is correct for finding the next index in linear probing?

Answer

-

Status : - *Marks : 0/1*

13. Which data structure is primarily used in linear probing?

Answer

-

Status : - *Marks : 0/1*

14. In the folding method, what is the primary reason for reversing alternate parts before addition?

Answer

-

Status : - *Marks : 0/1*

15. In division method, if key = 125 and m = 13, what is the hash index?

Answer

-

Status : - *Marks : 0/1*

16. In the division method of hashing, the hash function is typically written as:

Answer

-

Status : - *Marks : 0/1*

17. Which of the following values of 'm' is recommended for the division method in hashing?

Answer

-

Status : - *Marks : 0/1*

18. What is the output of the mid-square method for a key k = 123 if the hash table size is 10 and you extract the middle two digits of k * k?

Answer

-

Status : - ***Marks : 0/1***

19. In C, how do you calculate the mid-square hash index for a key k, assuming we extract two middle digits and the table size is 100?

Answer

-

Status : - ***Marks : 0/1***

20. What would be the result of folding 123456 into three parts and summing: (12 + 34 + 56)?

Answer

-

Status : - ***Marks : 0/1***

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Janani is a tech enthusiast who loves working with polynomials. She wants to create a program that can add polynomial coefficients and provide the sum of their coefficients.

The polynomials will be represented as a linked list, where each node of the linked list contains a coefficient and an exponent. The polynomial is represented in the standard form with descending order of exponents.

Input Format

The first line of input consists of an integer n, representing the number of terms in the first polynomial.

The following n lines of input consist of two integers each: the coefficient and the exponent of the term in the first polynomial.

The next line of input consists of an integer m, representing the number of terms in the second polynomial.

The following m lines of input consist of two integers each: the coefficient and the exponent of the term in the second polynomial.

Output Format

The output prints the sum of the coefficients of the polynomials.

Sample Test Case

Input: 3
2 2
3 1
4 0
3
2 2
3 1
4 0

Output: 18

Answer

-

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Arun is learning about data structures and algorithms. He needs your help in solving a specific problem related to a singly linked list.

Your task is to implement a program to delete a node at a given position. If the position is valid, the program should perform the deletion; otherwise, it should display an appropriate message.

Input Format

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

The second line consists of N space-separated elements of the linked list.

The third line consists of an integer x, representing the position to delete.

Position starts from 1.

Output Format

The output prints space-separated integers, representing the updated linked list after deleting the element at the given position.

If the position is not valid, print "Invalid position. Deletion not possible."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
8 2 3 1 7
2
Output: 8 3 1 7

Answer

-

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 1

Section 1 : Coding

1. Problem Statement

Imagine you are working on a text processing tool and need to implement a feature that allows users to insert characters at a specific position.

Implement a program that takes user inputs to create a singly linked list of characters and inserts a new character after a given index in the list.

Input Format

The first line of input consists of an integer N, representing the number of characters in the linked list.

The second line consists of a sequence of N characters, representing the linked list.

The third line consists of an integer index, representing the index(0-based) after which the new character node needs to be inserted.

The fourth line consists of a character value representing the character to be inserted after the given index.

Output Format

If the provided index is out of bounds (larger than the list size):

1. The first line of output prints "Invalid index".
2. The second line prints "Updated list: " followed by the unchanged linked list values.

Otherwise, the output prints "Updated list: " followed by the updated linked list after inserting the new character after the given index.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
a b c d e
2
X
Output: Updated list: a b c X d e

Answer

```
#include<stdio.h>
#include<stdlib.h>

typedef struct  Char {
   char value;
   struct Char* next;
} Node;
Node* newnode(char value) {
   Node* new_node = (Node*) malloc(sizeof(Node));
   new_node->value = value;
   new_node->next = NULL;
   return new_node;
```

```
}
 void insertNode(Node** head, char value) {
     Node* temp = *head;
     if(temp == NULL){
         *head = newnode(value);
         return;
    }
    while(temp->next != NULL){
        temp = temp->next;
    }
    temp->next = newnode(value);
 }
int length(Node* head) {
    int len = 0;
    while(head !=NULL) {
        head = head->next;
        len++;
    }
    return len;
}
void travers(Node* head){
    while(head != NULL){
        printf("%c ", head->value);
        head = head->next;
    }
    printf("\n");
}
 void insert(Node** head, int pos, char value) {
    if(pos >= length(* head)) {
        printf("Invalid index\n");
        return;
    }
    Node* temp = *head;
    for(int i = 0; i < pos; i++){
         temp = temp->next;
    }

    for(int i = 0; i < pos; i++){
        temp = temp->next;
    }
    Node* new_node = newnode(value);
    new_node-> next = temp->next;
    temp->next = new_node;
}
int main() {
    int n ;
    char value;
    Node* head = NULL;
    scanf("%d", &n);
    for(int i = 0; i <= n; i++) {
        scanf("%c ", &value);
        if(value == ' ' || value == '\n') {
            continue;
        }
        insertNode(&head, value);

    }
    scanf("%d %c", &n, &value);
    insert(&head, n, value);
    printf("Updated list: ");
    travers(head);
}
```

Status : Partially correct ***Marks : 1/10***

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

As part of a programming assignment in a data structures course, students are required to create a program to construct a singly linked list by inserting elements at the beginning.

You are an evaluator of the course and guide the students to complete the task.

Input Format

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

The second line consists of N space-separated integers.

Output Format

The output prints the singly linked list elements, after inserting them at the beginning.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
78 89 34 51 67

Output: 67 51 34 89 78

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
};

void insertAtFront(struct Node** head, int value) {
   struct Node* newn=(struct Node*)malloc(sizeof(struct Node));
   newn->data=value;
   newn->next=*head;
   *head=newn;
}
void printList(struct Node* head){
   struct Node* temp=head;
   while(temp!=NULL) {
      printf("%d ",temp->data);
      temp=temp->next;
   }
   printf("\n");
}

int main(){
   struct Node* head = NULL;

   int n;
   scanf("%d", &n);
```

```
    for (int i = 0; i < n; i++) {
        int activity;
        scanf("%d", &activity);
        insertAtFront(&head, activity);
    }

    printList(head);
    struct Node* current = head;
    while (current != NULL) {
        struct Node* temp = current;
        current = current->next;
        free(temp);
    }

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Imagine you are tasked with developing a simple GPA management system using a singly linked list. The system allows users to input student GPA values, insertion should happen at the front of the linked list, delete record by position, and display the updated list of student GPAs.

Input Format

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

The next n lines contain a single floating-point value representing the GPA of each student.

The last line contains an integer position, indicating the position at which a student record should be deleted. Position starts from 1.

Output Format

After deleting the data in the given position, display the output in the format "GPA: " followed by the GPA value, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
3.8
3.2
3.5
4.1
2
Output: GPA: 4.1
GPA: 3.2
GPA: 3.8

Answer

```
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
 struct node{
    float data;
    struct node *next;
};
void insert(struct node** head, float data) {
    struct node* newn=(struct node*)malloc(sizeof(struct node));
    newn->data=data;
    newn->next=*head;
    *head=newn;

}
void deletepos (struct node** head, int pos){
    struct node* temp=*head;
    if(pos==1){
       *head=temp->next;
        free(temp);
        return;
    }
    struct node* prev=NULL;
    for(int i=1;temp!=NULL && i<pos;i++){
        prev=temp;
        temp=temp->next;
    }
    prev->next=temp->next;
}
void print(struct node* head) {
    struct node* temp=head;
    while(temp!=NULL){
        printf("GPA: %.1f\n",temp->data);
        temp=temp->next;
    }
}
int main(){
    struct node* head=NULL;
    int n;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        float gpa;
        scanf("%f",&gpa);
        insert(&head,gpa);
    }
    int pos;
    scanf("%d",&pos);
    deletepos(&head,pos);
    print(head);
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 6

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John is tasked with creating a program to manage student roll numbers using a singly linked list.

Write a program for John that accepts students' roll numbers, inserts them at the end of the linked list, and displays the numbers.

Input Format

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

The second line consists of N space-separated integers, representing the roll numbers of students.

Output Format

The output prints the space-separated integers singly linked list, after inserting the roll numbers of students at the end.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
23 85 47 62 31

Output: 23 85 47 62 31

Answer

```
// You are using GCC
struct Node* insertAtEnd(struct Node* head,int rollNumber){
   struct Node* newn=(struct Node*)malloc(sizeof(struct Node));
   newn->rollNumber=rollNumber;
   newn->next=NULL;
   if(head==NULL) {
      head=newn;
      return newn;
   }
   struct Node* temp=head;
   while(temp->next!=NULL){
      temp=temp->next;
   }
   temp->next=newn;
   return head;
}
void display(struct Node* head){
   struct Node* temp=head;
   while(temp!=NULL){
      printf("%d ",temp->rollNumber);
      temp=temp->next;
   }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 7

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Dev is tasked with creating a program that efficiently finds the middle element of a linked list. The program should take user input to populate the linked list by inserting each element into the front of the list and then determining the middle element.

Assist Dev, as he needs to ensure that the middle element is accurately identified from the constructed singly linked list:

If it's an odd-length linked list, return the middle element.If it's an even-length linked list, return the second middle element of the two elements.

Input Format

The first line of input consists of an integer n, representing the number of elements in the linked list.

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

Output Format

The first line of output displays the linked list after inserting elements at the front.

The second line displays "Middle Element: " followed by the middle element of the linked list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 20 30 40 50

Output: 50 40 30 20 10
Middle Element: 30

Answer

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};


// You are using GCC
struct Node* push(struct Node* head,int data){
    struct Node* newn=(struct Node*)malloc(sizeof(struct Node));
    newn->data=data;
    newn->next=head;
    return newn;
}
int printMiddle(struct Node* head){
    struct Node* slow=head;
    struct Node* fast=head;
    while(fast!=NULL && fast->next!=NULL){
        slow=slow->next;
        fast=fast->next->next;
    }
    return slow->data;
}


int main() {
    struct Node* head = NULL;
    int n;

    scanf("%d", &n);
    int value;

    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = push(head, value);
    }

    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    printf("\n");


    int middle_element = printMiddle(head);
    printf("Middle Element: %d\n", middle_element);


    current = head;
    while (current != NULL) {
        struct Node* temp = current;
        current = current->next;
        free(temp);
    }

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Your task is to create a program to manage a playlist of items. Each item is represented as a character, and you need to implement the following operations on the playlist.

Here are the main functionalities of the program:

Insert Item: The program should allow users to add items to the front and end of the playlist. Items are represented as characters.Display Playlist: The program should display the playlist containing the items that were added.

To implement this program, a doubly linked list data structure should be used, where each node contains an item character.

Input Format

The input consists of a sequence of space-separated characters, representing the items to be inserted into the doubly linked list.

The input is terminated by entering - (hyphen).

Output Format

The first line of output prints "Forward Playlist: " followed by the linked list after inserting the items at the end.

The second line prints "Backward Playlist: " followed by the linked list after inserting the items at the front.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: a b c -

Output: Forward Playlist: a b c
Backward Playlist: c b a

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   char item;
   struct Node* next;
   struct Node* prev;
};

// You are using GCC
#include <iostream>
#include <string>
using namespace std;

struct Node {
   char data;
   Node* next;
   Node* prev;
   Node(char c) {
       data = c;
       next = NULL;
       prev = NULL;
   }
};

void insert_end(Node*& head, Node*& tail, char data) {
   Node* new_node = new Node(data);
   if (!head) {
       head = tail = new_node;
       return;
   }
   tail->next = new_node;
   new_node->prev = tail;
   tail = new_node;
}

void insert_front(Node*& head, Node*& tail, char data) {
   Node* new_node = new Node(data);
   if (!head) {
       head = tail = new_node;
       return;
   }
   head->prev = new_node;
   new_node->next = head;
   head = new_node;
}

void display_forward(Node* head) {
   Node* temp = head;
   while (temp) {
       cout << temp->data << " ";
       temp = temp->next;
   }
}

void display_backward(Node* tail) {
   Node* temp = tail;
   while (temp) {
       cout << temp->data << " ";
       temp = temp->prev;
   }
```

```
}

int main() {
    Node* forward_head = NULL;
    Node* forward_tail = NULL;
    Node* backward_head = NULL;
    Node* backward_tail = NULL;
    string item;
    while (cin >> item && item != "-") {
        char ch = item[0];
        insert_end(forward_head, forward_tail, ch);
        insert_front(backward_head, backward_tail, ch);
    }
    cout << "Forward Playlist: ";
    display_forward(forward_head);
    printf("\n");
    cout << " Backward Playlist: ";
    display_forward(backward_head);
    cout << endl;
    return 0;
}

int main() {
    struct Node* playlist = NULL;
    char item;

    while (1) {
        scanf(" %c", &item);
        if (item == '-') {
            break;
        }
        insertAtEnd(&playlist, item);
    }

    struct Node* tail = playlist;
    while (tail->next != NULL) {
        tail = tail->next;
    }

    printf("Forward Playlist: ");
    displayForward(playlist);

    printf("Backward Playlist: ");
    displayBackward(tail);

    freePlaylist(playlist);

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Moniksha, a chess coach organizing a tournament, needs a program to manage participant IDs efficiently. The program maintains a doubly linked list of IDs and offers two functions: Append to add IDs as students register, and Print Maximum ID to identify the highest ID for administrative tasks.

This tool streamlines tournament organization, allowing Moniksha to focus on coaching her students effectively.

Input Format

The first line consists of an integer n, representing the number of participant IDs to be added.

The second line consists of n space-separated integers representing the participant IDs.

Output Format

The output displays a single integer, representing the maximum participant ID.

If the list is empty, the output prints "Empty list!".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
163 137 155
Output: 163

Answer

```
# You are using Python
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
        self.prev = None

class DoublyLinkedList:
    def __init__(self):
        self.head = None
        self.tail = None

    def append(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = self.tail = new_node
            return
        self.tail.next = new_node
        new_node.prev = self.tail
        self.tail = new_node

    def find_max(self):
        if not self.head:
            return None
        max_id = self.head.data
        temp = self.head.next
        while temp:
            if temp.data > max_id:
                max_id = temp.data
            temp = temp.next
        return max_id

n_and_ids = input().strip().split()
if len(n_and_ids) == 1:
    n = int(n_and_ids[0])
    if n == 0:
        print("Empty list!")
    else:
        ids = list(map(int, input().strip().split()))
        dll = DoublyLinkedList()
        for id in ids:
            dll.append(id)
        print(dll.find_max())
else:
    n = int(n_and_ids[0])
    ids = list(map(int, n_and_ids[1:]))
    if n == 0:
        print("Empty list!")
    else:
        dll = DoublyLinkedList()
        for id in ids:
            dll.append(id)
        print(dll.find_max())
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Bob is tasked with developing a company's employee record management system. The system needs to maintain a list of employee records using a doubly linked list. Each employee is represented by a unique integer ID.

Help Bob to complete a program that adds employee records at the front, traverses the list, and prints the same for each addition of employees to the list.

Input Format

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

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

Output Format

For each employee ID, the program prints "Node Inserted" followed by the current state of the doubly linked list in the next line, with the data values of each node separated by spaces.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 4
101 102 103 104
Output: Node Inserted
101
Node Inserted
102 101
Node Inserted
103 102 101
Node Inserted
104 103 102 101
```

Answer

```
#include <iostream>
using namespace std;

struct node {
   int info;
   struct node* prev, * next;
};

struct node* start = NULL;

class DoublyLinkedList {
public:
   node* head;
   DoublyLinkedList() {
      head = NULL;
   }

   void insert_front(int data) {
      node* new_node = new node(data);
```

```
        if (head != NULL) {
            head->prev = new_node;
            new_node->next = head;
        }
        head = new_node;
    }

    void display() {
        node* temp = head;
        while (temp != NULL) {
            cout << temp->data << " ";
            temp = temp->next;
        }
    }
};


int main() {
    int n, data;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> data;
        insertAtFront(data);
        traverse();
    }
    return 0;
}
```

Status : Wrong ***Marks : 0/10***

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ravi is developing a student registration system for a college. To efficiently store and manage the student IDs, he decides to implement a doubly linked list where each node represents a student's ID.

In this system, each student's ID is stored sequentially, and the system needs to display all registered student IDs in the order they were entered.

Implement a program that creates a doubly linked list, inserts student IDs, and displays them in the same order.

Input Format

The first line contains an integer N the number of student IDs.

The second line contains N space-separated integers representing the student IDs.

Output Format

The output should display the single line containing N space-separated integers representing the student IDs stored in the doubly linked list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 20 30 40 50
Output: 10 20 30 40 50

Answer

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* prev;
   struct Node* next;
};

struct Node* createNode(int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = data;
   newNode->prev = NULL;
   newNode->next = NULL;
   return newNode;
}

void append(struct Node** head, struct Node** tail, int data) {
   struct Node* newNode = createNode(data);
   if (*head == NULL) {
      *head = *tail = newNode;
   } else {
      (*tail)->next = newNode;
```

```
        newNode->prev = *tail;
        *tail = newNode;
    }
}

void display(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
}

int main() {
    int n, val;
    scanf("%d", &n);
    struct Node* head = NULL;
    struct Node* tail = NULL;
    for (int i = 0; i < n; i++) {
        scanf("%d", &val);
        append(&head, &tail, val);
    }
    display(head);
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ashwin is tasked with developing a simple application to manage a list of items in a shop inventory using a doubly linked list. Each item in the inventory has a unique identification number. The application should allow users to perform the following operations:

Create a List of Items: Initialize the inventory with a given number of items. Each item will be assigned a unique number provided by the user and insert the elements at end of the list.

Delete an Item: Remove an item from the inventory at a specific position.

Display the Inventory: Show the list of items before and after deletion.

If the position provided for deletion is invalid (e.g., out of range), it should display an error message.

Input Format

The first line contains an integer n, representing the number of items to be initially entered into the inventory.

The second line contains n integers, each representing the unique identification number of an item separated by spaces.

The third line contains an integer p, representing the position of the item to be deleted from the inventory.

Output Format

The first line of output prints "Data entered in the list:" followed by the data values of each node in the doubly linked list before deletion.

If p is an invalid position, the output prints "Invalid position. Try again."

If p is a valid position, the output prints "After deletion the new list:" followed by the data values of each node in the doubly linked list after deletion.

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 4
1 2 3 4
5
Output: Data entered in the list:
 node 1 : 1
 node 2 : 2
 node 3 : 3
 node 4 : 4
Invalid position. Try again.
```

Answer

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>
```

```
struct Node {
    int data;
    struct Node* prev;
    struct Node* next;
};

struct Node* head = NULL;

void insertEnd(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (head == NULL) {
        newNode->prev = NULL;
        head = newNode;
    } else {
        struct Node* temp = head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
        newNode->prev = temp;
    }
}

void displayList() {
    struct Node* temp = head;
    int count = 1;
    while (temp != NULL) {
        printf("node %d : %d\n", count, temp->data);
        temp = temp->next;
        count++;
    }
}

int getLength() {
    int length = 0;
    struct Node* temp = head;
    while (temp != NULL) {
        length++;
        temp = temp->next;
    }
    return length;
}

void displayAfterDeletion() {
    printf("After deletion the new list:\n");
    displayList();
}

void deleteAtPosition(int pos) {
    if (pos <= 0 || pos > getLength()) {
        printf("Invalid position. Try again.\n");
        return;
    }

    struct Node* temp = head;

    if (pos == 1) {
        head = head->next;
        if (head != NULL)
            head->prev = NULL;
        free(temp);
        displayAfterDeletion();
        return;
    }

    for (int i = 1; i < pos; i++)
        temp = temp->next;

    if (temp->prev != NULL)
        temp->prev->next = temp->next;
    if (temp->next != NULL)
        temp->next->prev = temp->prev;

    free(temp);
    displayAfterDeletion();
}

int main() {
    int n, p, val;

    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &val);
        insertEnd(val);
    }

    scanf("%d", &p);

    printf("Data entered in the list:\n");
    displayList();

    deleteAtPosition(p);

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Imagine a bustling coffee shop, where customers are placing their orders for their favorite coffee drinks. The cafe owner Sheeren wants to efficiently manage the queue of coffee orders using a digital system. She needs a program to handle this queue of orders.

You are tasked with creating a program that implements a queue for coffee orders. Each character in the queue represents a customer's coffee order, with 'L' indicating a latte, 'E' indicating an espresso, 'M' indicating a macchiato, 'O' indicating an iced coffee, and 'N' indicating a nabob.

Customers can place orders and enjoy their delicious coffee drinks.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Enqueue the coffee order into the queue. If the choice is 1, the following input is a space-separated character ('L', 'E', 'M', 'O', 'N').

Choice 2: Dequeue a coffee order from the queue.

Choice 3: Display the orders in the queue.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the queue:

If the choice is 1:

1. Insert the given order into the queue and display "Order for [order] is enqueued." where [order] is the coffee order that is inserted.
2. If the queue is full, print "Queue is full. Cannot enqueue more orders."

If the choice is 2:

1. Dequeue a character from the queue and display "Dequeued Order: " followed by the corresponding order that is dequeued.
2. If the queue is empty without any orders, print "No orders in the queue."

If the choice is 3:

1. The output prints "Orders in the queue are: " followed by the space-separated orders present in the queue.
2. If there are no orders in the queue, print "Queue is empty. No orders available."

If the choice is 4:

1. Exit the program and print "Exiting program"

If any other choice is entered, the output prints "Invalid option."

Refer to the sample output for the exact text and format.

Sample Test Case

```
Input: 1 L
1 E
1 M
1 O
1 N
1 O
3
2
3
4
Output: Order for L is enqueued.
Order for E is enqueued.
Order for M is enqueued.
Order for O is enqueued.
Order for N is enqueued.
Queue is full. Cannot enqueue more orders.
Orders in the queue are: L E M O N
Dequeued Order: L
Orders in the queue are: E M O N
Exiting program
```

Answer

```
#include <stdio.h>
#define MAX_SIZE 5

char orders[MAX_SIZE];
int front = -1;
int rear = -1;

void initializeQueue() {
   front = -1;
   rear = -1;
}
// You are using GCC
```

```c
#include <stdio.h>

#define SIZE 5

char queue[SIZE];
int front = -1, rear = -1;
void enqueue(char order) {
    if (rear == SIZE - 1) {
        printf("Queue is full. Cannot enqueue more orders.\n");
        return;
    }
    if (front == -1) front = 0;
    queue[++rear] = order;
    printf("Order for %c is enqueued.\n", order);
}


void dequeue() {
    if (front == -1 || front > rear) {
        printf("No orders in the queue.\n");
        return;
    }
    printf("Dequeued Order: %c\n", queue[front++]);
}


void displayQueue() {
    if (front == -1 || front > rear) {
        printf("Queue is empty. No orders available.\n");
        return;
    }
    printf("Orders in the queue are: ");
    for (int i = front; i <= rear; i++) {
        printf("%c ", queue[i]);
    }
    printf("\n");
}


int main() {
    int choice;
    char order;
    while (1) {
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                scanf(" %c", &order);
                enqueue(order);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                displayQueue();
                break;
            case 4:
                printf("Exiting program\n");
                return 0;
            default:
                printf("Invalid option.\n");
        }
    }
}

int main() {
    char order;
    int option;
    initializeQueue();
    while (1) {
        if (scanf("%d", &option) != 1) {
            break;
        }
        switch (option) {
            case 1:
                if (scanf(" %c", &order) != 1) {
                    break;
                }
                if (enqueue(order)) {
                }
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program");
                return 0;
            default:
                printf("Invalid option.\n");
                break;
        }
    }
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

In a bustling IT department, staff regularly submit helpdesk tickets to request technical assistance. Managing these tickets efficiently is vital for providing quality support.

Your task is to develop a program that uses an array-based queue to handle and prioritize helpdesk tickets based on their unique IDs.

Implement a program that provides the following functionalities:

Enqueue Helpdesk Ticket: Add a new helpdesk ticket to the end of the queue. Provide a positive integer representing the ticket ID for the new ticket.Dequeue Helpdesk Ticket: Remove and process the next helpdesk ticket from the front of the queue. The program will display the ticket ID of the processed ticket.Display Queue: Display the ticket IDs of all the helpdesk tickets currently in the queue.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Enqueue the ticket ID into the queue. If the choice is 1, the following input is a space-separated integer, representing the ticket ID to be enqueued into the queue.

Choice 2: Dequeue a ticket from the queue.

Choice 3: Display the ticket IDs in the queue.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the queue:

If the choice is 1:

1. Insert the given ticket ID into the queue and display "Helpdesk Ticket ID [id] is enqueued." where [id] is the ticket ID that is inserted.
2. If the queue is full, print "Queue is full. Cannot enqueue."

If the choice is 2:

1. Dequeue a ticket ID from the queue and display "Dequeued Helpdesk Ticket ID: " followed by the corresponding ID that is dequeued.
2. If the queue is empty without any elements, print "Queue is empty."

If the choice is 3:

1. The output prints "Helpdesk Ticket IDs in the queue are: " followed by the space-separated ticket IDs present in the queue.
2. If there are no elements in the queue, print "Queue is empty."

If the choice is 4:

1. Exit the program and print "Exiting the program"

If any other choice is entered, print "Invalid option."

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 1 101
1 202
1 203
1 204
1 205
1 206
3
2
3
4
```

```
Output: Helpdesk Ticket ID 101 is enqueued.
Helpdesk Ticket ID 202 is enqueued.
Helpdesk Ticket ID 203 is enqueued.
Helpdesk Ticket ID 204 is enqueued.
Helpdesk Ticket ID 205 is enqueued.
Queue is full. Cannot enqueue.
Helpdesk Ticket IDs in the queue are: 101 202 203 204 205
Dequeued Helpdesk Ticket ID: 101
Helpdesk Ticket IDs in the queue are: 202 203 204 205
Exiting the program
```

Answer

```c
#include <stdio.h>
#define MAX_SIZE 5

int ticketIDs[MAX_SIZE];
int front = -1;
int rear = -1;
int lastDequeued;

void initializeQueue() {
   front = -1;
   rear = -1;
}
```

```
#include <iostream>
using namespace std;

#define MAX_SIZE 5

int queue[MAX_SIZE];
int front = -1, rear = -1;

bool isFull() {
    return (front == 0 && rear == MAX_SIZE - 1) || (rear + 1 == front);
}

bool isEmpty() {
    return front == -1;
}

void enqueue(int id) {
    if (isFull()) {
        cout << "Queue is full. Cannot enqueue." << endl;
        return;
    }
    if (front == -1) {
        front = rear = 0;
    } else if (rear == MAX_SIZE - 1 && front != 0) {
        rear = 0;
    } else {
        rear++;
    }
    queue[rear] = id;
    cout << "Helpdesk Ticket ID " << id << " is enqueued." << endl;
}

void dequeue() {
    if (isEmpty()) {
        cout << "Queue is empty." << endl;
        return;
    }
    int id = queue[front];
    if (front == rear) {
        front = rear = -1;
    } else if (front == MAX_SIZE - 1) {
        front = 0;
    } else {
        front++;
    }
    cout << "Dequeued Helpdesk Ticket ID: " << id << endl;
}

void display() {
    if (isEmpty()) {
        cout << "Queue is empty." << endl;
        return;
    }
    cout << "Helpdesk Ticket IDs in the queue are: ";
    int i = front;
    while (true) {
        cout << queue[i] << " ";
        if (i == rear)
            break;
        i = (i + 1) % MAX_SIZE;
    }
    cout << endl;
}

int main() {
    int choice, val;
    while (true) {
        cin >> choice;
        switch (choice) {
            case 1:
                cin >> val;
                enqueue(val);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                cout << "Exiting the program" << endl;
                return 0;
            default:
                cout << "Invalid option." << endl;
        }
    }
    return 0;
}

int main() {
    int ticketID;
    int option;
    initializeQueue();
    while (1) {
        if (scanf("%d", &option) == EOF) {
            break;
        }
        switch (option) {
            case 1:
                if (scanf("%d", &ticketID) == EOF) {
                    break;
                }
                enqueue(ticketID);
                break;
            case 2:
                if (dequeue()) {
                    printf("Dequeued Helpdesk Ticket ID: %d\n", lastDequeued);
                } else {
                    printf("Queue is empty.\n");
                }
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting the program\n");
                return 0;
            default:
                printf("Invalid option.\n");
                break;
        }
    }
    return 0;
}
```

Status : Wrong ***Marks : 0/10***

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Write a program to implement a queue using an array and pointers. The program should provide the following functionalities:

Insert an element into the queue.Delete an element from the queue.Display the elements in the queue.

The queue has a maximum capacity of 5 elements. If the queue is full and an insertion is attempted, a "Queue is full" message should be displayed. If the queue is empty and a deletion is attempted, a "Queue is empty" message should be displayed.

Input Format

Each line contains an integer representing the chosen option from 1 to 3.

Option 1: Insert an element into the queue followed by an integer representing the element to be inserted, separated by a space.

Option 2: Delete an element from the queue.

Option 3: Display the elements in the queue.

Output Format

For option 1 (insertion):-

1. The program outputs: "<data> is inserted in the queue." if the data is successfully inserted.
2. "Queue is full." if the queue is already full and cannot accept more elements.

For option 2 (deletion):-

1. The program outputs: "Deleted number is: <data>" if an element is successfully deleted and returns the value of the deleted element.
2. "Queue is empty." if the queue is empty no elements can be deleted.

For option 3 (display):-

1. The program outputs: "Elements in the queue are: <element1> <element2> ... <elementN>" where <element1>, <element2>, ..., <elementN> represent the elements present in the queue.
2. "Queue is empty." if the queue is empty no elements can be displayed.

For invalid options, the program outputs: "Invalid option."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 10

3
5
Output: 10 is inserted in the queue.
Elements in the queue are: 10
Invalid option.

Answer

-

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In an office setting, a print job management system is used to efficiently handle and process print jobs. The system is implemented using a queue data structure with an array.

The program provides the following operations:

Enqueue Print Job: Add a print job with a specified number of pages to the end of the queue.Dequeue Print Job: Remove and process the next print job in the queue.Display Queue: Display the print jobs in the queue

The program should ensure that print jobs are processed in the order they are received.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Enqueue the print job into the queue. If the choice is 1, the following input is a space-separated integer, representing the pages to be enqueued into the queue.

Choice 2: Dequeue a print job from the queue.

Choice 3: Display the print jobs in the queue.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the queue:

If the choice is 1:

1. Insert the given page into the queue and display "Print job with [page] pages is enqueued." where [page] is the number of pages that are inserted.
2. If the queue is full, print "Queue is full. Cannot enqueue."

If the choice is 2:

1. Dequeue a page from the queue and display "Processing print job: [page] pages" where [page] is the corresponding page that is dequeued.
2. If the queue is empty without any elements, print "Queue is empty."

If the choice is 3:

1. The output prints "Print jobs in the queue: " followed by the space-separated pages present in the queue.
2. If there are no elements in the queue, print "Queue is empty."

If the choice is 4:

1. Exit the program and print "Exiting program"

If any other choice is entered, the output prints "Invalid option."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1
10
1
20
1
30
1
40
1
50
1
60
3
2
3
4
Output: Print job with 10 pages is enqueued.
Print job with 20 pages is enqueued.
Print job with 30 pages is enqueued.
Print job with 40 pages is enqueued.
Print job with 50 pages is enqueued.
Queue is full. Cannot enqueue.
Print jobs in the queue: 10 20 30 40 50
Processing print job: 10 pages
Print jobs in the queue: 20 30 40 50
Exiting program

Answer

```
#include <stdio.h>

#define MAX_SIZE 5

int queue[MAX_SIZE];
int front = -1, rear = -1;
```

```
void enqueue(int pages) {
    if ((front == 0 && rear == MAX_SIZE - 1) || (rear + 1 == front)) {
        printf("Queue is full. Cannot enqueue.\n");
        return;
    }
    if (front == -1) {
        front = rear = 0;
    } else if (rear == MAX_SIZE - 1 && front != 0) {
        rear = 0;
    } else {
        rear++;
    }
    queue[rear] = pages;
    printf("Print job with %d pages is enqueued.\n", pages);
}

void dequeue() {
    if (front == -1) {
        printf("Queue is empty.\n");
        return;
    }
    int pages = queue[front];
    if (front == rear) {
        front = rear = -1;
    } else if (front == MAX_SIZE - 1) {
        front = 0;
    } else {
        front++;
    }
    printf("Processing print job: %d pages\n", pages);
}

void display() {
    if (front == -1) {
        printf("Queue is empty.\n");
        return;
    }
    printf("Print jobs in the queue: ");
    int i = front;
    while (true) {
        printf("%d ", queue[i]);
        if (i == rear)
            break;
        i = (i + 1) % MAX_SIZE;
    }
    printf("\n");
}

int main() {
    int choice, pages;
    while (1) {
        if (scanf("%d", &choice) != 1)
            break;
        switch (choice) {
            case 1:
                if (scanf("%d", &pages) == 1) {
                    enqueue(pages);
                }
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program\n");
                return 0;
            default:
                printf("Invalid option.\n");
        }
    }
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are tasked with implementing basic operations on a queue data structure using a linked list.

You need to write a program that performs the following operations on a queue:

Enqueue Operation: Implement a function that inserts an integer element at the rear end of the queue.Print Front and Rear: Implement a function that prints the front and rear elements of the queue. Dequeue Operation: Implement a function that removes the front element from the queue.

Input Format

The first line of input consists of an integer N, representing the number of elements to be inserted into the queue.

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

Output Format

The first line prints "Front: X, Rear: Y" where X is the front and Y is the rear elements of the queue.

The second line prints the message indicating that the dequeue operation (front element removed) is performed: "Performing Dequeue Operation:".

The last line prints "Front: M, Rear: N" where M is the front and N is the rear elements after the dequeue operation.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
12 56 87 23 45

Output: Front: 12, Rear: 45
Performing Dequeue Operation:
Front: 56, Rear: 45

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
};

struct Node* front = NULL;
struct Node* rear = NULL;

// You are using GCC
#include <stdio.h>
#include <stdlib.h>
```

```
struct Node {
    int data;
    struct Node* next;
};

struct Node* front = NULL;
struct Node* rear = NULL;

void enqueue(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    if (rear == NULL) {
        front = rear = newNode;
    } else {
        rear->next = newNode;
        rear = newNode;
    }
}

void printFrontAndRear() {
    if (front != NULL && rear != NULL) {
        printf("Front: %d, Rear: %d\n", front->data, rear->data);
    }
}

void dequeue() {
    if (front == NULL)
        return;
    struct Node* temp = front;
    front = front->next;
    if (front == NULL)
        rear = NULL;
    free(temp);
}

int main() {
    int N;
    scanf("%d", &N);
    for (int i = 0; i < N; i++) {
        int val;
        scanf("%d", &val);
        enqueue(val);
    }
    printFrontAndRear();
    printf("Performing Dequeue Operation:\n");
    dequeue();
    printFrontAndRear();
    return 0;
}

int main() {
    int n, data;
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &data);
        enqueue(data);
    }
    printFrontRear();
    printf("Performing Dequeue Operation:\n");
    dequeue();
    printFrontRear();
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John is learning about Binary Search Trees (BST) in his computer science class. He wants to create a program that allows users to delete a node with a given value from a BST and print the remaining nodes using an in-order traversal.

Implement a function to help him delete a node with a given value from a BST.

Input Format

The first line of input consists of an integer N, representing the number of nodes in the BST.

The second line consists of N space-separated integers, representing the values of the BST nodes.

The third line consists of an integer V, which is the value to delete from the BST.

Output Format

The output prints the space-separated values in the BST in an in-order traversal, after the deletion of the specified value.

If the specified value is not available in the tree, print the given input values in-order traversal.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 5 15 2 7
15

Output: 2 5 7 10

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct TreeNode {
    int data;
    struct TreeNode* left;
    struct TreeNode* right;
};

struct TreeNode* createNode(int key) {
    struct TreeNode* newNode = (struct TreeNode*)malloc(sizeof(struct
TreeNode));
    newNode->data = key;
    newNode->left = newNode->right = NULL;
    return newNode;
}
struct TreeNode* insert(struct TreeNode* root, int key) {
    if (!root) return createNode(key);
    if (key < root->data) root->left = insert(root->left, key);
    else root->right = insert(root->right, key);
    return root;
}

struct TreeNode* findMin(struct TreeNode* root) {
    while (root->left) root = root->left;
    return root;
}

struct TreeNode* deleteNode(struct TreeNode* root, int key) {
    if (!root) return NULL;
    if (key < root->data) root->left = deleteNode(root->left, key);
    else if (key > root->data) root->right = deleteNode(root->right, key);
    else {
        if (!root->left) {
            struct TreeNode* temp = root->right;
            free(root);
            return temp;
        } else if (!root->right) {
            struct TreeNode* temp = root->left;
            free(root);
            return temp;
        }
        struct TreeNode* temp = findMin(root->right);
        root->data = temp->data;
        root->right = deleteNode(root->right, temp->data);
    }
    return root;
}

void inorderTraversal(struct TreeNode* root) {
    if (root) {
        inorderTraversal(root->left);
        printf("%d ", root->data);
        inorderTraversal(root->right);
    }
}

int main()
{
    int N, rootValue, V;
    scanf("%d", &N);
    struct TreeNode* root = NULL;
    for (int i = 0; i < N; i++) {
        int key;
        scanf("%d", &key);
        if (i == 0) rootValue = key;
        root = insert(root, key);
    }
    scanf("%d", &V);
    root = deleteNode(root, V);
    inorderTraversal(root);
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Mike is learning about Binary Search Trees (BSTs) and wants to implement various operations on them. He wants to write a basic program for creating a BST, inserting nodes, and printing the tree in the pre-order traversal.

Write a program to help him solve this program.

Input Format

The first line of input consists of an integer N, representing the number of values to insert into the BST.

The second line consists of N space-separated integers, representing the values to insert into the BST.

Output Format

The output prints the space-separated values of the BST in the pre-order traversal.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
3 1 5 2 4

Output: 3 1 2 5 4

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* left;
   struct Node* right;
};

struct Node* createNode(int value) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = value;
   newNode->left = newNode->right = NULL;
   return newNode;
}
struct Node* insert(struct Node* root, int value) {
   if (!root) return createNode(value);
   if (value < root->data) root->left = insert(root->left, value);
   else root->right = insert(root->right, value);
   return root;
}

void printPreorder(struct Node* root) {
   if (root) {
      printf("%d ", root->data);
      printPreorder(root->left);
      printPreorder(root->right);
```

```
    }
}
int main() {
    struct Node* root = NULL;

    int n;
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        int value;
        scanf("%d", &value);
        root = insert(root, value);
    }

    printPreorder(root);
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are required to implement basic operations on a Binary Search Tree (BST), like insertion and searching.

Insertion: Given a list of integers, construct a Binary Search Tree by repeatedly inserting each integer into the tree according to the rules of a BST.

Searching: Given an integer, search for its presence in the constructed Binary Search Tree. Print whether the integer is found or not.

Write a program to calculate this efficiently.

Input Format

The first line of input consists of an integer n, representing the number of nodes in the binary search tree.

The second line consists of the values of the nodes, separated by space as integers.

The third line consists of an integer representing, the value that is to be searched.

Output Format

The output prints, "Value <value> is found in the tree." if the given value is present, otherwise it prints: "Value <value> is not found in the tree."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7
8 3 10 1 6 14 23
6

Output: Value 6 is found in the tree.

Answer

```
struct Node* insertNode(struct Node* root, int value) {
   if (root == NULL) return createNode(value);
   if (value < root->data) root->left = insertNode(root->left, value);
   else if (value > root->data) root->right = insertNode(root->right, value);
   return root;
}

struct Node* searchNode(struct Node* root, int key) {
   if (root == NULL || root->data == key) return root;
   return key < root->data ? searchNode(root->left, key) : searchNode(root->right,
key);
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John, a computer science student, is learning about binary search trees (BST) and their properties. He decides to write a program to create a BST, display it in post-order traversal, and find the minimum value present in the tree.

Help him by implementing the program.

Input Format

The first line of input consists of an integer N, representing the number of elements to insert into the BST.

The second line consists of N space-separated integers data, which is the data to be inserted into the BST.

Output Format

The first line of output prints the space-separated elements of the BST in post-order traversal.

The second line prints the minimum value found in the BST.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
5 10 15
Output: 15 10 5
The minimum value in the BST is: 5

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* left;
   struct Node* right;
};

struct Node* createNode(int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = data;
   newNode->left = newNode->right = NULL;
   return newNode;
}

struct Node* insert(struct Node* root, int data) {
   if (root == NULL) return createNode(data);
   if (data < root->data) root->left = insert(root->left, data);
   else if (data > root->data) root->right = insert(root->right, data);
   return root;
}

void displayTreePostOrder(struct Node* root) {
   if (root == NULL) return;
   displayTreePostOrder(root->left);
   displayTreePostOrder(root->right);
   printf("%d ", root->data);
}

int findMinValue(struct Node* root) {
   while (root->left != NULL) root = root->left;
   return root->data;
}

int main() {
   struct Node* root = NULL;
   int n, data;
   scanf("%d", &n);

   for (int i = 0; i < n; i++) {
       scanf("%d", &data);
       root = insert(root, data);
   }

   displayTreePostOrder(root);
   printf("\n");

   int minValue = findMinValue(root);
   printf("The minimum value in the BST is: %d", minValue);

   return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In his computer science class, John is learning about Binary Search Trees (BST). He wants to build a BST and find the maximum value in the tree.

Help him by writing a program to insert nodes into a BST and find the maximum value in the tree.

Input Format

The first line of input consists of an integer N, representing the number of nodes in the BST.

The second line consists of N space-separated integers, representing the values of the nodes to insert into the BST.

Output Format

The output prints the maximum value in the BST.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 5 15 2 7

Output: 15

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct TreeNode {
   int data;
   struct TreeNode* left;
   struct TreeNode* right;
};
struct TreeNode* createNode(int key) {
   struct TreeNode* newNode = (struct TreeNode*)malloc(sizeof(struct TreeNode));
   newNode->data = key;
   newNode->left = newNode->right = NULL;
   return newNode;
}
struct TreeNode* insert(struct TreeNode* root, int key) {
   if (root == NULL) return createNode(key);
   if (key < root->data) root->left = insert(root->left, key);
   else if (key > root->data) root->right = insert(root->right, key);
   return root;
}

int findMax(struct TreeNode* root) {
   if (root == NULL) return -1;
   while (root->right != NULL) root = root->right;
   return root->data;
}
int main() {
   int N, rootValue;
   scanf("%d", &N);

   struct TreeNode* root = NULL;

   for (int i = 0; i < N; i++) {
       int key;
       scanf("%d", &key);
       if (i == 0) rootValue = key;
       root = insert(root, key);
   }

   int maxVal = findMax(root);
   if (maxVal != -1) {
       printf("%d", maxVal);
   }

   return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John and Mary are collaborating on a project that involves data analysis. They each have a set of age data, one sorted in ascending order and the other in descending order. However, their analysis requires the data to be in ascending order.

Write a program to help them merge the two sets of age data into a single sorted array in ascending order using merge sort.

Input Format

The first line of input consists of an integer N, representing the number of age values in each dataset.

The second line consists of N space-separated integers, representing the ages of participants in John's dataset (in ascending order).

The third line consists of N space-separated integers, representing the ages of participants in Mary's dataset (in descending order).

Output Format

The output prints a single line containing space-separated integers, which represents the merged dataset of ages sorted in ascending order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
1 3 5 7 9
10 8 6 4 2

Output: 1 2 3 4 5 6 7 8 9 10

Answer

```
#include <stdio.h>

// You are using GCC
#include <iostream>
#include <vector>
#include <algorithm>

void merge_sorted_arrays(int n, std::vector<int>& asc_list, std::vector<int>& 
desc_list) {
   std::reverse(desc_list.begin(), desc_list.end());
   std::vector<int> merged_list = asc_list;
   merged_list.insert(merged_list.end(), desc_list.begin(), desc_list.end());
   std::sort(merged_list.begin(), merged_list.end());

   for (int age : merged_list) {
      std::cout << age << " ";
   }
   std::cout << std::endl;
}

int main() {
   int n;
```

```
    std::cin >> n;

    std::vector<int> asc_list(n), desc_list(n);
    for (int i = 0; i < n; i++) {
        std::cin >> asc_list[i];
    }
    for (int i = 0; i < n; i++) {
        std::cin >> desc_list[i];
    }

    merge_sorted_arrays(n, asc_list, desc_list);

    return 0;
}
int main() {
    int n, m;
    scanf("%d", &n);
    int arr1[n], arr2[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr1[i]);
    }
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr2[i]);
    }
    int merged[n + n];
    mergeSort(arr1, n);
    mergeSort(arr2, n);
    merge(merged, arr1, arr2, n, n);
    for (int i = 0; i < n + n; i++) {
        printf("%d ", merged[i]);
    }
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Nandhini asked her students to arrange a set of numbers in ascending order. She asked the students to arrange the elements using insertion sort, which involves taking each element and placing it in its appropriate position within the sorted portion of the array.

Assist them in the task.

Input Format

The first line of input consists of the value of n, representing the number of array elements.

The second line consists of n elements, separated by a space.

Output Format

The output prints the sorted array, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
67 28 92 37 59

Output: 28 37 59 67 92

Answer

```
#include <stdio.h>

// You are using GCC
#include <iostream>
#include <vector>

void insertion_sort(std::vector<int>& arr, int n) {
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}

int main() {
    int n;
    std::cin >> n;

    std::vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        std::cin >> arr[i];
    }

    insertion_sort(arr, n);
    for (int i = 0; i < n; i++) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}

int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    insertionSort(arr, n);
    printArray(arr, n);
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are the lead developer of a text-processing application that assists writers in organizing their thoughts. One crucial feature is a character-sorting service that helps users highlight the most critical elements of their text.

To achieve this, you decide to enhance the service to sort characters in descending order using the Quick-Sort algorithm. Implement the algorithm to efficiently rearrange the characters, ensuring that it is sorted in descending order.

Input Format

The first line of the input consists of a positive integer value N, representing the number of characters to be sorted.

The second line of input consists of N space-separated lowercase alphabetical characters.

Output Format

The output displays the set of alphabetical characters, sorted in descending order.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
a d g j k

Output: k j g d a

Answer

```
#include <stdio.h>
#include <string.h>

// You are using GCC
#include <iostream>
#include <vector>
#include <algorithm>

void quick_sort(std::vector<char>& arr, int low, int high) {
   if (low < high) {
      char pivot = arr[high];
      int i = low - 1;

      for (int j = low; j < high; j++) {
         if (arr[j] > pivot) {
            i++;
            std::swap(arr[i], arr[j]);
         }
      }
      std::swap(arr[i + 1], arr[high]);

      int pi = i + 1;
      quick_sort(arr, low, pi - 1);
```

```
        quick_sort(arr, pi + 1, high);
    }
}

int main() {
    int n;
    std::cin >> n;

    std::vector<char> arr(n);
    for (int i = 0; i < n; i++) {
        std::cin >> arr[i];
    }

    quick_sort(arr, 0, n - 1);

    for (char ch : arr) {
        std::cout << ch << " ";
    }
    std::cout << std::endl;

    return 0;
}
int main() {
    int n;
    scanf("%d", &n);

    char characters[n];

    for (int i = 0; i < n; i++) {
        char input;
        scanf(" %c", &input);
        characters[i] = input;
    }

    quicksort(characters, 0, n - 1);

    for (int i = 0; i < n; i++) {
        printf("%c ", characters[i]);
    }

    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Kavya, a software developer, is analyzing data trends. She has a list of integers and wants to identify the nth largest number in the list after sorting the array using QuickSort.

To optimize performance, Kavya is required to use QuickSort to sort the list before finding the nth largest number.

Input Format

The first line of input consists of an integer n, representing the size of the array.

The second line consists of n space-separated integers, representing the elements of the array nums.

The third line consists of an integer k, representing the position of the largest number you need to print after sorting the array.

Output Format

The output prints the k-th largest number in the sorted array (sorted in ascending order).

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 6
-1 0 1 2 -1 -4
3
Output: 0

Answer

```
#include <stdio.h>
#include <stdlib.h>

#include <stdio.h>

// Function to swap two elements
void swap(int* a, int* b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}

// Partition function for QuickSort (Lomuto partition scheme)
int partition(int arr[], int low, int high) {
   int pivot = arr[high];
   int i = low - 1; // index of smaller element

   for (int j = low; j <= high - 1; j++) {
      if (arr[j] <= pivot) {
         i++;
         swap(&arr[i], &arr[j]);
      }
   }
```

```
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}

// QuickSort recursive function
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    int k;
    scanf("%d", &k);

    // Sort the array using QuickSort
    quickSort(arr, 0, n - 1);

    // Print the k-th largest element
    // Since array is sorted ascending, k-th largest is arr[n-k]
    printf("%d\n", arr[n - k]);

    return 0;
}
int main() {
    int n, k;
    scanf("%d", &n);
    int* nums = (int*)malloc(n * sizeof(int));
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }
    scanf("%d", &k);
    findNthLargest(nums, n, k);
    free(nums);
    return 0;
}
```

Status : Wrong ***Marks : 0/10***

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 6_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Jose has an array of N fractional values, represented as double-point numbers. He needs to sort these fractions in increasing order and seeks your help.

Write a program to help Jose sort the array using the merge sort algorithm.

Input Format

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

The second line consists of N double-point numbers, separated by spaces, representing the fractions array.

Output Format

The output prints N double-point numbers, sorted in increasing order, and rounded to three decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
0.123 0.543 0.321 0.789

Output: 0.123 0.321 0.543 0.789

Answer

```
#include <stdio.h>
#include <stdlib.h>

// You are using GCC
#include <iostream>
#include <vector>
#include <iomanip>

void merge(std::vector<double>& arr, int left, int mid, int right) {
   int n1 = mid - left + 1, n2 = right - mid;
   std::vector<double> leftArr(n1), rightArr(n2);

   for (int i = 0; i < n1; i++) leftArr[i] = arr[left + i];
   for (int i = 0; i < n2; i++) rightArr[i] = arr[mid + 1 + i];

   int i = 0, j = 0, k = left;
   while (i < n1 && j < n2) {
      if (leftArr[i] <= rightArr[j]) arr[k++] = leftArr[i++];
      else arr[k++] = rightArr[j++];
   }

   while (i < n1) arr[k++] = leftArr[i++];
   while (j < n2) arr[k++] = rightArr[j++];
}

void merge_sort(std::vector<double>& arr, int left, int right) {
   if (left < right) {
      int mid = left + (right - left) / 2;
```

```
        merge_sort(arr, left, mid);
        merge_sort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

int main() {
    int n;
    std::cin >> n;

    std::vector<double> arr(n);
    for (int i = 0; i < n; i++) {
        std::cin >> arr[i];
    }

    merge_sort(arr, 0, n - 1);

    for (double num : arr) {
        std::cout << std::fixed << std::setprecision(3) << num << " ";
    }
    std::cout << std::endl;

    return 0;
}
int main() {
    int n;
    scanf("%d", &n);
    double fractions[n];
    for (int i = 0; i < n; i++) {
        scanf("%lf", &fractions[i]);
    }
    mergeSort(fractions, 0, n - 1);
    for (int i = 0; i < n; i++) {
        printf("%.3f ", fractions[i]);
    }
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ravi is building a basic hash table to manage student roll numbers for quick lookup. He decides to use Linear Probing to handle collisions.

Implement a hash table using linear probing where:

The hash function is: index = roll_number % table_sizeOn collision, check subsequent indexes (i+1, i+2, ...) until an empty slot is found.

You need to:

Insert a list of n student roll numbers into the hash table.Print the final state of the hash table. If a slot is empty, print -1.

Input Format

The first line of the input contains two integers n and table_size, where n is the number of roll numbers to be inserted, and table_size is the size of the hash table.

The second line contains n space-separated integers — the roll numbers to insert into the hash table.

Output Format

The output should print a single line with table_size space-separated integers representing the final state of the hash table after all insertions.

If any slot remains unoccupied, it should be represented as -1.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4 7
50 700 76 85

Output: 700 50 85 -1 -1 -1 76

Answer

```
#include <stdio.h>

#define MAX 100


// Initialize the hash table with -1 (indicating empty slots)
void initializeTable(int table[], int size) {
    for (int i = 0; i < size; i++) {
        table[i] = -1;
    }
}

// Function to find an empty slot using linear probing
int linearProbe(int table[], int size, int num) {
    int index = num % size;

    // Keep checking the next slot until an empty space is found
    while (table[index] != -1) {
        index = (index + 1) % size; // Move to the next slot (wrap around if needed)
    }

    return index;
}

// Insert elements into the hash table using linear probing
void insertIntoHashTable(int table[], int size, int arr[], int n) {
    for (int i = 0; i < n; i++) {
        int index = linearProbe(table, size, arr[i]);
        table[index] = arr[i];
    }
}

// Print the final state of the hash table
void printTable(int table[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", table[i]);
    }
    printf("\n");
}


int main() {
    int n, table_size;
    scanf("%d %d", &n, &table_size);

    int arr[MAX];
    int table[MAX];

    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    initializeTable(table, table_size);
    insertIntoHashTable(table, table_size, arr, n);
    printTable(table, table_size);

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Priya is developing a simple student management system. She wants to store roll numbers in a hash table using Linear Probing, and later search for specific roll numbers to check if they exist.

Implement a hash table using linear probing with the following operations:

Insert all roll numbers into the hash table.For a list of query roll numbers, print "Value x: Found" or "Value x: Not Found" depending on whether it exists in the table.

Input Format

The first line contains two integers, n and table_size — the number of roll numbers to insert and the size of the hash table.

The second line contains n space-separated integers — the roll numbers to insert.

The third line contains an integer q — the number of queries.

The fourth line contains q space-separated integers — the roll numbers to search for.

Output Format

The output print q lines — for each query value x, print: "Value x: Found" or "Value x: Not Found"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5 10
21 31 41 51 61
3
31 60 51

Output: Value 31: Found
Value 60: Not Found
Value 51: Found

Answer

```
#include <stdio.h>

#define MAX 100



// Initialize the hash table with -1 (indicating empty slots)
void initializeTable(int table[], int size) {
    for (int i = 0; i < size; i++) {
        table[i] = -1;
    }
}

// Function to find an empty slot using linear probing
int linearProbe(int table[], int size, int num) {
    int index = num % size;

    // Keep checking the next slot until an empty space is found
    while (table[index] != -1) {
        index = (index + 1) % size; // Move to the next slot (wrap around if needed)
    }

    return index;
}

// Insert elements into the hash table using linear probing
void insertIntoHashTable(int table[], int size, int arr[], int n) {
    for (int i = 0; i < n; i++) {
        int index = linearProbe(table, size, arr[i]);
        table[index] = arr[i];
    }
}

// Search for a roll number in the hash table
int searchInHashTable(int table[], int size, int num) {
    int index = num % size;

    // Check slots using linear probing
    while (table[index] != -1) {
        if (table[index] == num)
            return 1; // Found
        index = (index + 1) % size; // Move to the next slot
    }

    return 0; // Not Found
}



int main() {
    int n, table_size;
    scanf("%d %d", &n, &table_size);

    int arr[MAX], table[MAX];
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    initializeTable(table, table_size);
    insertIntoHashTable(table, table_size, arr, n);

    int q, x;
    scanf("%d", &q);
    for (int i = 0; i < q; i++) {
        scanf("%d", &x);
        if (searchInHashTable(table, table_size, x))
            printf("Value %d: Found\n", x);
        else
            printf("Value %d: Not Found\n", x);
    }

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a messaging application, users maintain a contact list with names and corresponding phone numbers. Develop a program to manage this contact list using a dictionary implemented with hashing.

The program allows users to add contacts, delete contacts, and check if a specific contact exists. Additionally, it provides an option to print the contact list in the order of insertion.

Input Format

The first line consists of an integer n, representing the number of contact pairs to be inserted.

Each of the next n lines consists of two strings separated by a space: the name of the contact (key) and the corresponding phone number (value).

The last line contains a string k, representing the contact to be checked or removed.

Output Format

If the given contact exists in the dictionary:

1. The first line prints "The given key is removed!" after removing it.
2. The next n - 1 lines print the updated contact list in the format: "Key: X; Value: Y" where X represents the contact's name and Y represents the phone number.

If the given contact does not exist in the dictionary:

1. The first line prints "The given key is not found!".
2. The next n lines print the original contact list in the format: "Key: X; Value: Y" where X represents the contact's name and Y represents the phone number.

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: 3
Alice 1234567890
Bob 9876543210
Charlie 4567890123
Bob

Output: The given key is removed!
Key: Alice; Value: 1234567890
Key: Charlie; Value: 4567890123

Answer

```
// You are using GCC
#include <iostream>
#include <unordered_map>
#include <vector>
using namespace std;
```

```
int main() {
    int n;
    cin >> n;

    unordered_map<string, string> contacts;
    vector<pair<string, string>> contact_list;

    // Insert contacts
    for (int i = 0; i < n; i++) {
        string name, phone;
        cin >> name >> phone;
        contacts[name] = phone;
        contact_list.push_back({name, phone}); // Maintain insertion order
    }

    // Search or remove contact
    string key;
    cin >> key;

    auto it = contacts.find(key);
    if (it != contacts.end()) {
        // Contact found, remove it
        contacts.erase(key);
        cout << "The given key is removed!" << endl;

        // Print updated contact list
        for (const auto &entry : contact_list) {
            if (entry.first != key) {
                cout << "Key: " << entry.first << "; Value: " << entry.second << endl;
            }
        }
    } else {
        // Contact not found
        cout << "The given key is not found!" << endl;
        for (const auto &entry : contact_list) {
            cout << "Key: " << entry.first << "; Value: " << entry.second << endl;
        }
    }

    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Develop a program using hashing to manage a fruit contest where each fruit is assigned a unique name and a corresponding score. The program should allow the organizer to input the number of fruits and their names with scores.

Then, it should enable them to check if a specific fruit, identified by its name, is part of the contest. If the fruit is registered, the program should display its score; otherwise, it should indicate that it is not included in the contest.

Input Format

The first line consists of an integer N, representing the number of fruits in the contest.

The following N lines contain a string K and an integer V, separated by a space, representing the name and score of each fruit in the contest.

The last line consists of a string T, representing the name of the fruit to search for.

Output Format

If T exists in the dictionary, print "Key "T" exists in the dictionary.".

If T does not exist in the dictionary, print "Key "T" does not exist in the dictionary.".

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: 2
banana 2
apple 1
Banana

Output: Key "Banana" does not exist in the dictionary.

Answer

```
// You are using GCC
#include <iostream>
#include <unordered_map>
using namespace std;

int main() {
    int N;
    cin >> N;

    unordered_map<string, int> fruitScores;

    // Read fruit names and scores
    for (int i = 0; i < N; i++) {
        string fruit;
        int score;
        cin >> fruit >> score;
        fruitScores[fruit] = score; // Store in hash table
    }

    // Read the fruit to be searched
    string T;
    cin >> T;

    // Check if the fruit exists in the contest
    if (fruitScores.find(T) != fruitScores.end()) {
        cout << "Key \"" << T << "\" exists in the dictionary." << endl;
    } else {
        cout << "Key \"" << T << "\" does not exist in the dictionary." << endl;
    }

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are provided with a collection of numbers, each represented by an array of integers. However, there's a unique scenario: within this array, one element occurs an odd number of times, while all other elements occur an even number of times. Your objective is to identify and return the element that occurs an odd number of times in this arrangement.

Utilize mid-square hashing by squaring elements and extracting middle digits for hash codes. Implement a hash table for efficient integer occurrence tracking.

Note: Hash function: squared = key * key.

Example

Input:

7

2 2 3 3 4 4 5

Output:

5

Explanation

The hash function and the calculated hash indices for each element are as follows:

2 -> hash(2*2) % 100 = 4

3 -> hash(3*3) % 100 = 9

4 -> hash(4*4) % 100 = 16

5 -> hash(5*5) % 100 = 25

The hash table records the occurrence of each element's hash index:

Index 4: 2 occurrences

Index 9: 2 occurrences

Index 16: 2 occurrences

Index 25: 1 occurrence

Among the elements, the integer 5 occurs an odd number of times (1 occurrence) and satisfies the condition of the problem. Therefore, the program outputs 5.

Input Format

The first line of input consists of an integer N, representing the size of the array.

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

Output Format

The output prints a single integer representing the element that occurs an odd number of times.

If no such element exists, print -1.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7
2 2 3 3 4 4 5

Output: 5

Answer

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>

#define MAX_SIZE 100

// You are using GCC
#include <iostream>
#include <unordered_map>
using namespace std;

int findOddOccurrence(int arr[], int N) {
   unordered_map<int, int> hashTable;

   // Compute hash index using mid-square hashing
   for (int i = 0; i < N; i++) {
      int hashIndex = (arr[i] * arr[i]) % 100;  // Mid-square hashing
      hashTable[hashIndex]++;  // Count occurrences
   }

   // Identify the element occurring an odd number of times
   for (int i = 0; i < N; i++) {
      int hashIndex = (arr[i] * arr[i]) % 100;
      if (hashTable[hashIndex] % 2 == 1) {
         return arr[i];  // Return the element occurring an odd number of times
      }
```

```
    }
    return -1;  // If no odd occurrence exists
}

int main() {
    int N;
    cin >> N;
    int arr[N];

    for (int i = 0; i < N; i++) {
        cin >> arr[i];
    }

    cout << findOddOccurrence(arr, N) << endl;

    return 0;
}
int main() {
    int n;
    scanf("%d", &n);

    int arr[MAX_SIZE];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("%d\n", getOddOccurrence(arr, n));

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: kishore k
Email: 240701267@rajalakshmi.edu.in
Roll no: 240701267
Phone: 8122803620
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 14

Section 1 : MCQ

1. What will be the value of the following Python expression?

4 + 3 % 5

Answer

7

Status : Correct *Marks : 1/1*

2. Which of the following represents the bitwise XOR operator?

Answer

^

Status : Correct *Marks : 1/1*

3. The value of the expressions 4/(3*(2-1)) and 4/3*(2-1) is the same. True or False?

Answer

True

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

4. Evaluate the expression given below if A= 16 and B = 15

A % B // A

Answer

0

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

5. What is used to concatenate two strings in Python?

Answer

+ operator

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

6. What is the output of the following program?

print((1, 2) + (3, 4))

Answer

(1, 2, 3, 4)

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

7. Which of the following operators has its associativity from right to left?

Answer

**

Status : Correct *Marks : 1/1*

8. Which is the correct operator for power(xy)?

Answer

x**y

Status : Correct *Marks : 1/1*

9. What is the output of the following number conversion?

```
z = complex(1.25)
print(z)
```

Answer

(1.25+0j)

Status : Correct *Marks : 1/1*

10. Which of the following is an example of the type casting?

Answer

float(5)

Status : Wrong *Marks : 0/1*

11. What is the output of the below expression?

```
print(3*1**3)
```

Answer

3

Status : Correct *Marks : 1/1*

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

```
x = int(34.56 - 2 * 2)
print(x)
```

Answer

30

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

13. Which of these is not a core data type?

Answer

Class

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

14. What will be the output for the below code?

```
x=15
y=12
print(x&y)
```

Answer

12

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

15. Which of the following functions converts a string to a float in Python?

Answer

float(x)

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ram is working as a developer for BrightEdu Coaching Center, which wants to build a student fee management system.

Each student’s enrollment has:

An Enrollment ID (integer)A Student Name (string)The Number of Subjects (integer)

The fee calculation rules are:

Registration Fee = 1000 units (flat for every student).Per Subject Fee = 800 units.If the student enrolls in more than 5 subjects, a 20% scholarship (discount) is applied on the total fee.

Ram has been asked to implement this system using:

A class with attributes for student details.A constructor to initialize student details.Setter methods to update details if needed.Getter methods to retrieve details.Objects of the class to represent student enrollments.

Finally, display each student's details and final fee.

Input Format

The first line of input contains an integer N, representing the number of students.

For each student:

- The next line contains the Enrollment ID (integer).
- The following line contains the student's name (string).
- The next line contains the Number of subjects (integer).

Output Format

For each student, print the details in the following format:

- Enrollment ID: <enrollment_id>
- Student Name: <student_name>
- Final Fee: <final_fee> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234
Ravi Kumar
3

Output: Enrollment ID: 1234
Student Name: Ravi Kumar
Final Fee: 3400.0

Answer

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

class Student {
```

```
private int enrollmentId;
private String studentName;
private int numberOfSubjects;

public Student(int enrollmentId, String studentName, int numberOfSubjects) {
    this.enrollmentId = enrollmentId;
    this.studentName = studentName;
    this.numberOfSubjects = numberOfSubjects;
}

public void setEnrollmentId(int enrollmentId) {
    this.enrollmentId = enrollmentId;
}

public void setStudentName(String studentName) {
    this.studentName = studentName;
}

public void setNumberOfSubjects(int numberOfSubjects) {
    this.numberOfSubjects = numberOfSubjects;
}

public int getEnrollmentId() {
    return enrollmentId;
}

public String getStudentName() {
    return studentName;
}

public int getNumberOfSubjects() {
    return numberOfSubjects;
}

public double calculateFee() {
    double registrationFee = 1000;
    double perSubjectFee = 800;
    double totalFee = registrationFee + (numberOfSubjects * perSubjectFee);

    if (numberOfSubjects > 5) {
        totalFee = totalFee * 0.8;
    }
```

```
        return Math.round(totalFee * 10) / 10.0;
    }
}

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

        int N = Integer.parseInt(sc.nextLine().trim());

        for (int i = 0; i < N; i++) {
            int enrollmentId = Integer.parseInt(sc.nextLine().trim());
            String studentName = sc.nextLine().trim();
            int numberOfSubjects = Integer.parseInt(sc.nextLine().trim());

            Student student = new Student(enrollmentId, studentName,
numberOfSubjects);

            System.out.println("Enrollment ID: " + student.getEnrollmentId());
            System.out.println("Student Name: " + student.getStudentName());
            System.out.println("Final Fee: " + student.calculateFee());
        }

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
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 sc = new Scanner(System.in);


    int N = sc.nextInt();
    sc.nextLine();


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


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

```
        String searchName = sc.next();


        int count = 0;
        for (String name : names) {
            if (name.equals(searchName)) {
                count++;
            }
        }


        System.out.println(count);

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 7_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 15

Section 1 : MCQ

1. Which of the following statements about Java interfaces is true?

Answer

A class can implement multiple interfaces.

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

2. Which of the following is the correct way to declare an interface in Java?

Answer

interface Vehicle {   void start();}

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

3. What happens when an implementing class does not override a default method from an interface?

Answer

The default method's implementation from the interface will be used.

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

4. What is the output of the following code?

```
interface A {
   default void show() {
      System.out.println("A's Default Method");
   }
}

class B {
   public void show() {
      System.out.println("B's Method");
   }
}

class C extends B implements A {
}

public class Main {
   public static void main(String[] args) {
      C obj = new C();
      obj.show();
   }
}
```

Answer

B's Method

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

5. What is the output of the following code?

```
interface A {
   static void display() {
      System.out.println("Static method in A");
   }
}

class B implements A {
   static void display() {
      System.out.println("Static method in B");
   }
}

public class Main {
   public static void main(String[] args) {
      B.display();
   }
}
```

Answer

Static method in B

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

6. What is the output of the following code?

```
interface X {
   default void show() {
      System.out.println("X's Default Method");
   }
}

interface Y {
   default void show() {
      System.out.println("Y's Default Method");
   }
}

class Z implements X, Y {
   public void show() {
      System.out.println("Z's Method");
   }
}

public class Main {
   public static void main(String[] args) {
      Z obj = new Z();
      obj.show();
   }
}
```

Answer

Z's Method

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

7. If a class implements two interfaces that have the same default method, what must the class do?

Answer

The class must override the method to resolve ambiguity.

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

8. Consider a class implementing an interface and extending a class, both having a method with the same name. Which method gets called?

Answer

The method from the superclass

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

9. How can a class explicitly call a default method from an interface if there is a naming conflict?

Answer

Using InterfaceName.super.methodName();

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

10. Which of the following statements is true regarding default methods in Java interfaces?

Answer

A default method can be overridden in a class implementing the interface.

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

11. Can a Java interface contain both default and static methods?

Answer

Yes, an interface can have both default and static methods.

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

12. What is the primary purpose of static methods in Java interfaces?

Answer

They allow an interface to provide helper methods without requiring an implementing class.

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

13. How do you call a static method from an interface MyInterface?

Answer

MyInterface.staticMethod();

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

14. What is the output of the following code?

```
interface MathOperations {
    static int square(int x) {
        return x * x;
    }
}

public class Main {
    public static void main(String[] args) {
        System.out.println(MathOperations.square(5));
    }
}
```

Answer

25

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

15. What is the output of the following code?

```
interface A {
    default void show() {
        System.out.println("A's Default Method");
    }
}

interface B {
    default void show() {
        System.out.println("B's Default Method");
    }
}

class C implements A, B {
    public void show() {
        A.super.show();
    }
}

public class Main {
    public static void main(String[] args) {
        C obj = new C();
        obj.show();
    }
}
```

Answer

A's Default Method

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

Rajalakshmi Engineering College

Scan to verify results

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 9_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 10

Section 1 : MCQ

1. 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*

2. What is Collection in Java?

Answer

A group of objects

Status : Correct *Marks : 1/1*

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

Answer

list.get(0);

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("apple");
        list.add("banana");
        list.add("cherry");
        list.add("banana");
        System.out.println(list.lastIndexOf("banana"));
    }
}
```

Answer

4

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

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

Answer

removeFirst()

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

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

Answer

Adds an element to the beginning of the list

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(1);
      list.add(2);
      list.add(3);
      list.add(4);
      list.set(2, 10);
      System.out.println(list);
   }
}
```

Answer

[1, 2, 3, 4]

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

8. 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***

9. 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, 20, 30]

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

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

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

Answer

30

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<String> list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        list.remove("Apple");
        System.out.println(list);

    }
}
```

Answer

[Banana]

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

13. 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***

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

Answer

push()

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

15. 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

1

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 15

Section 1 : MCQ

1. Which of the following data types is used to store single characters?

Answer

char

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

2. What is the output of the following code?

```
import java.util.*;

class RelationalOperatorExample {
   public static void main(String[] args) {
      int x = 8, y = 4;
      boolean result = (x != y);
```

```
        System.out.println(result);
    }
}
```

Answer

true

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

3. Which of the following is not a primitive data type?

Answer

string

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

4. What is the output of the following program?

```
class Arithmetic {
    public static void main(String[] args) {
        char ch = 'A';
        System.out.println(ch);
    }
}
```

Answer

A

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

5. What is the output of the following code?

```
class TestClass {
    public static void main(String[] args) {
        int count = 8;
        count = count ^ 1;
        System.out.println(count);
    }
}
```

Answer

9

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

6. What is the output of the following code?

```
class TestClass {
    public static void main(String[] args) {
        int a = 10;
        int b = 3;
        System.out.println(a / b);
    }
}
```

Answer

3

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

7. What is the output of the following code?

```
class TestClass {
    public static void main(String[] args) {
        int a = 5;
        int b = 10;

        int sum = a + b;
        int bitwiseAnd = a & b;
        int bitwiseOr = a | b;

        System.out.println(sum);
        System.out.println(bitwiseAnd);
        System.out.println(bitwiseOr);
    }
}
```

Answer

15015

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

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

```
class DivisionExample {
    public static void main(String[] args) {
        double num1 = 10.5;
        double num2 = 3;
        int result = (int)(num1 / num2);
        System.out.println(result);
    }
}
```

Answer

3

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

9. What is the result of the following expression?

```
import java.util.*;

class ComplexExpressionExample {
    public static void main(String[] args) {
        int a = 5, b = 2, c = 3, d = 4;
        int result = a + b * c / d - b;

        System.out.println(result);
    }
}
```

Answer

4

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

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

```
class DataTypesMCQ {
   public static void main(String[] args) {
      int a = 10;
      double b = 5;
      System.out.println(a / b);
   }
}
```

Answer

2.0

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

11. Which of the following data types is used to store floating-point numbers with greater precision?

Answer

double

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

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

```
import java.util.*;

class TernaryOperatorExample {
   public static void main(String[] args) {
      int a = 5, b = 10;
      int result = (a > b) ? a : b;
      System.out.println(result);
   }
}
```

Answer

10

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

13. What will be the output of the following code snippet?

```
import java.util.*;

class OperatorPrecedenceExample {
   public static void main(String[] args) {
      int a = 5, b = 3, c = 2;
      int result = a + b * c;

      System.out.println(result);
   }
}
```

Answer

11

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

14. What is the output of the following code?

```
class TestClass {
   public static void main(String[] args) {
      int x = 5;
      int X = 10;

      int sum = x + X;
      int bitwiseResult = x | X;

      System.out.println(sum);
      System.out.println(bitwiseResult);
   }
}
```

Answer

1515

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

15. What is the output of the following program?

```
class Demo {
   public static void main(String[] args) {
      String text = "Hello, World!";
      System.out.println(text);
   }
}
```

Answer

Hello, World!

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

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

```
class Person {
   String name;
   void setName(String n) {
      name = n;
   }
   void printName() {
      System.out.println(name);
   }
}

class Test {
   public static void main(String[] args) {
      Person p = new Person();
      p.printName();
```

```
    }
}
```

Answer

null

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

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

```
class Ball {
    int size = 11;
}

class Game {
    public static void main(String[] args) {
        Ball b1 = new Ball();
        Ball b2 = new Ball();
        b2.size = 10;
        System.out.println(b1.size);
    }
}
```

Answer

11

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

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

```
class Box {
    int volume(int l, int b, int h) {
        return l * b * h;
    }
}

public class Main {
    public static void main(String[] args) {
        Box b = new Box();
        System.out.println(b.volume(2, 3, 4));
    }
}
```

Answer

24

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

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

```
class A {
    int y = 30;
}

public class Main {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new A();
        a1.y = 50;
        System.out.println(a2.y);
    }
}
```

Answer

30

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

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

```
class Demo {
    void printMessage() {
        System.out.println("Hello from Demo");
    }
}

public class Main {
    public static void main(String[] args) {
        Demo d = new Demo();
        d.printMessage();
    }
}
```

Answer

Hello from Demo

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

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

```
class MathUtils {
    int add(int x) {
        return x + x;
    }
}

public class Main {
    public static void main(String[] args) {
        MathUtils m = new MathUtils();
        System.out.println(m.add(5));
    }
}
```

Answer

10

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

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

```
class Test {
    private int value;
    Test(int value) {
        this.value = value;
    }
    public int getValue() {
        return value;
    }
}
public class Main {
    public static void main(String[] args) {
        Test obj = new Test(10);
        System.out.println(obj.value);
    }
}
```

Answer

Compile-time error

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

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

```
class Person {
    int age = 18;
}

public class Main {
    public static void main(String[] args) {
        Person p = new Person();
        p.age += 2;
        System.out.println("Age: " + p.age);
    }
}
```

Answer

Age: 20

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

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

```
class A {
    int val = 20;
}
```

```
public class Main {
    public static void main(String[] args) {
        A obj1 = new A();
        A obj2 = obj1;
        obj2.val += 5;
        System.out.println(obj1.val);
    }
}
```

Answer

5

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

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

```
class Sample {
    int x = 10;

    void display() {
        System.out.println("x = " + x);
    }

    public static void main(String[] args) {
        Sample s = new Sample();
        s.display();
    }
}
```

Answer

x = 10

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

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

```
class A {
    int x = 50;
}
```

```
public class Main {
   public static void main(String[] args) {
      A obj1 = new A();
      A obj2 = obj1;
      obj2.x = 100;
      System.out.println(obj1.x);
   }
}
```

Answer

100

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

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

```
class Box {
   int length = 5;
   int width = 4;

   int area() {
      return length * width;
   }

   public static void main(String[] args) {
      Box b = new Box();
      System.out.println("Area = " + b.area());
   }
}
```

Answer

Compilation error

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

13. What is the output of the following code?

```
class Box {
    int height;
    Box(int height) {
        this.height = height;
    }
    void modifyHeight(Box b) {
        b.height += 10;
    }
}
public class Main {
    public static void main(String[] args) {
        Box b1 = new Box(20);
        b1.modifyHeight(b1);
        System.out.println(b1.height);
    }
}
```

Answer

30

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

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

```
class Alpha {
    void greet(String name) {
        System.out.println("Hello " + name);
    }
}

public class Main {
    public static void main(String[] args) {
        Alpha obj = new Alpha();
        obj.greet("Anu");
    }
}
```

Answer

Hello Anu

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

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

```
class A {
   int p = 5;
   int q = 2;
}

class Main {
   public static void main(String[] args) {
      A obj = new A();
      System.out.println(obj.p + obj.q);
   }
}
```

Answer

7

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

Rajalakshmi Engineering College

Name: kishore v
Email: 240701270@rajalakshmi.edu.in
Roll no: 240701270
Phone: 9003716759
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 8_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 15

Section 1 : MCQ

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

```
class MyException extends Exception {
   public MyException() {
      super("Default Exception Message");
   }
}

class Test {
   public static void main(String[] args) {
      try {
         throw new MyException();
      } catch (MyException e) {
         System.out.println(e.getMessage());
      }
   }
}
```

Answer

Default Exception Message

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

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

```
import java.io.*;

class NegativeAgeException extends Exception {
   public NegativeAgeException(String message) {
      super(message);
   }
}

class Test {
   public static void main(String[] args) {
      try {
         int age = -5;
         if (age < 0) {
            throw new NegativeAgeException("Age cannot be negative");
         }
      } catch (NegativeAgeException e) {
         System.out.println(e.getMessage());
      }
   }
}
```

Answer

Age cannot be negative

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

3. What will be the output for the following code?

```
import java.io.*;
```

```
class TemperatureTooHighException extends Exception {
   public TemperatureTooHighException(String message) {
      super(message);
   }
}

class Test {
   public static void main(String[] args) {
      try {
         int temperature = 110;
         if (temperature > 100) {
            throw new TemperatureTooHighException("Temperature too
high");
         }
      } catch (TemperatureTooHighException e) {
         System.out.println(e.getMessage());
      }
   }
}
```

Answer

Temperature too high

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

4. what is the output of the following code?

```
class MyException extends Exception {
   public MyException(String message) {
      super(message);
   }
}

class Test {
   public static void main(String[] args) {
      try {
         throw new MyException("Error occurred");
      } catch (MyException e) {
         System.out.println(e);
```

```
        }
    }
}
```

Answer

MyException: Error occurred

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

5. What is the purpose of a custom exception in Java?

Answer

To create user-defined exceptions for specific scenarios

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

6. How do you create an unchecked custom exception?

Answer

By extending RuntimeException

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

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

```
class InvalidUsernameException extends Exception {
    public InvalidUsernameException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            String username = "abc";
            if (username.length() < 5) {
                throw new InvalidUsernameException("Username must be at
least 5 characters long");
```

```
            }
        } catch (InvalidUsernameException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

Username must be at least 5 characters long

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

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

```
class NegativeBalanceException extends Exception {
    public NegativeBalanceException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            double balance = -500;
            if (balance < 0) {
                throw new NegativeBalanceException("Balance cannot be
negative");
            }
        } catch (NegativeBalanceException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Answer

Error: Balance cannot be negative

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

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

```
class InvalidVotingAgeException extends Exception {
    public InvalidVotingAgeException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int age = 15;
            if (age < 18) {
                throw new InvalidVotingAgeException("You are not eligible to
vote");
            }
            System.out.println("Eligible to vote");
        } catch (InvalidVotingAgeException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

You are not eligible to vote

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

10. Which keyword is used to explicitly throw a custom exception?

Answer

throw

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

11. what is the output of the following code?

```
class MyException extends Exception {
    public MyException(String message) {
        super(message);
    }
}

class Test {
    static void check() throws MyException {
        throw new MyException("Custom Exception Occurred");
    }

    public static void main(String[] args) {
        try {
            check();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

Custom Exception Occurred

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

12. What will be the output for the following code?

```
import java.io.*;

class OutOfStockException extends Exception {
    public OutOfStockException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int stock = 0;
            if (stock == 0) {
                throw new OutOfStockException("Item is out of stock");
            }
        } catch (OutOfStockException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

Item is out of stock

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

13. What will happen if a checked custom exception is thrown inside a method without being caught or declared?

Answer

Compilation Error

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

14. What will be the output for the following code?

```
import java.io.*;

class UnderageException extends Exception {
    public UnderageException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int age = 17;
            if (age < 18) {
                throw new UnderageException("Underage, cannot proceed");
            }
        } catch (UnderageException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

Underage, cannot proceed

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

15. Which of the following is true about custom exceptions?

Answer

Custom exceptions must extend either Exception or RuntimeException

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