

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

- What will be the output of the following code?

```
class Test {  
    public static void main(String[] args) {  
        int a = 4, b = 5;  
        if ((a + b) % 2 == 0)  
            System.out.print("Even");  
        else  
            System.out.print("Odd");  
    }  
}
```

Answer

Odd

Status : Correct

Marks : 1/1

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

```
class ConditionTest {  
    public static void main(String[] args) {  
        int x = 10;  
        if (x > 5)  
            System.out.print("High");  
    }  
}
```

Answer

High

Status : Correct

Marks : 1/1

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

```
class Loop {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 3; i++) {  
            for (int j = 1; j <= 2; j++) {  
                System.out.print(i + " " + j + " ");  
            }  
        }  
    }  
}
```

Answer

11 12 21 22 31 32

Status : Correct

Marks : 1/1

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

```
public class Main {  
    public static void main(String[] args) {  
        int sum = 0;  
        for(int i = 1; i <= 5; i++) {  
            sum += i;  
        }  
    }  
}
```

```
        }
    }
}
```

Answer

15

Status : Correct

Marks : 1/1

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

```
class Test {
    public static void main(String[] args) {
        int x = 5, y = 2;
        if (x + y == 10)
            System.out.print("Ten");
        else if (x - y == 3)
            System.out.print("Three");
        else
            System.out.print("None");
    }
}
```

Answer

Three

Status : Correct

Marks : 1/1

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

```
public class Main {
    public static void main(String[] args) {
        for(int i = 1; i <= 20; i = i * 2) {
            System.out.print(i + " ");
        }
    }
}
```

Answer

1 2 4 8 16

Status : Correct

Marks : 1/1

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

```
class ConditionTest {  
    public static void main(String[] args) {  
        int a = 7;  
        if (a == 7)  
            System.out.print("Match");  
        else  
            System.out.print("No Match");  
    }  
}
```

Answer

Match

Status : Correct

Marks : 1/1

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

```
public class Main {  
    public static void main(String[] args) {  
        int i = 10;  
        do {  
            System.out.print(i + " ");  
            i -= 3;  
        } while(i > 0);  
    }  
}
```

Answer

10 7 4 1

Status : Correct

Marks : 1/1

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

```
public class Main {  
    public static void main(String[] args) {  
        int day = 4;  
        String result = "";  
        switch(day) {  
            case 1:  
                result = "Monday";  
                break;  
            case 2:  
                result = "Tuesday";  
                break;  
            case 3:  
                result = "Wednesday";  
                break;  
            default:  
                result = "Other Day";  
        }  
        System.out.println(result);  
    }  
}
```

Answer

Other Day

Status : Correct

Marks : 1/1

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

```
class LoopTest {  
    public static void main(String[] args) {  
        int i = 1;  
        do {  
            System.out.print(i + " ");  
            i *= 2;  
        } while (i <= 8);  
    }  
}
```

Answer

1 2 4 8

Status : Correct

Marks : 1/1

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

```
class LoopTest {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i > 0) {  
            System.out.print(i + " ");  
            i++;  
            if (i == 5) break;  
        }  
    }  
}
```

Answer

1 2 3 4 5

Status : Wrong

Marks : 0/1

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

```
class Test {  
    public static void main(String[] args) {  
        int num = 15;  
        if (num > 10)  
            if (num % 3 == 0)  
                System.out.print("Divisible");  
            else  
                System.out.print("Not Divisible");  
    }  
}
```

Answer

Divisible

Status : Correct

Marks : 1/1

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

```
public class Main {  
    public static void main(String[] args) {  
        int i = 1;  
        while(i < 10) {  
            i += 2;  
        }  
        System.out.println(i);  
    }  
}
```

Answer

10

Status : Wrong

Marks : 0/1

14. What will be the output of the following Java code snippet?

```
public class Main {  
    public static void main(String[] args) {  
        int score = 75;  
        if(score >= 90) {  
            System.out.println("Grade: A");  
        } else if(score >= 80) {  
            System.out.println("Grade: B");  
        } else if(score >= 70) {  
            System.out.println("Grade: C");  
        } else {  
            System.out.println("Grade: D");  
        }  
    }  
}
```

Answer

Grade: C

Status : Correct

Marks : 1/1

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

```
class Main {  
    public static void main(String[] args) {  
        for (int i = 5; i > 0; i--) {  
            System.out.print(i + " ");  
        }  
    }  
}
```

Answer

5 4 3 2 1

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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;
public class Main{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int a = input.nextInt();
        int b = input.nextInt();
        int c = input.nextInt();
        int d = input.nextInt();
        int e = input.nextInt();
        int avg = (a+b+c+d+e)/5;
        System.out.println("Average score: " + avg);
        if(avg>50){
            System.out.println("The student has passed");
        }
        else{
            System.out.println("The student has failed");
        }
        input.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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.Scanner;
public class Main{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        if(n%5==0){
            System.out.println(n + " is a multiple of 5");
        }
        else if(n%7==0){
            System.out.println(n + " is a multiple of 7");
        }
        else{
            System.out.println(n + " is neither multiple of 5 nor 7");
        }
        input.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



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

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

public class Main {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        double height=sc.nextDouble();
        double weight=sc.nextDouble();
        double bmi=weight/(height*height);
        System.out.printf("BMI: %.2f ",bmi);
        if(bmi<18.5)
            System.out.print("Classification: Underweight");
        else if(bmi>=18.6 && bmi<=24.9)
            System.out.print("Classification: Normal Weight");
        else if(bmi>=25.0 && bmi<=29.9)
            System.out.print("Classification: Overweight");
        else
```

```
        System.out.print("Classification: Obese");  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_PAH

Attempt : 2
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Sampad is a high school teacher who wants to convert numeric grades into letter grades.

Write a program that accepts a numeric grade as input. The program should then convert this numeric grade into a letter grade based on specific conditions. The letter grades are A, B, C, D and F.

The conversion is determined by the following conditions:

If the numeric grade is 90 or higher, it's an "A"
If the numeric grade is between 80 and 89 (inclusive), it's a "B"
If the numeric grade is between 70 and 79 (inclusive), it's a "C"
If the numeric grade is between 60 and 69 (inclusive), it's a "D"
If the numeric grade is below 60, it's an "F"

Input Format

The input consists of an integer representing the numeric grade of the student.

Output Format

The output prints the letter grade corresponding to the input numeric grade as "Letter Grade: <grade>".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 85

Output: Letter Grade: B

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 grade = sc.nextInt();

        if (grade >= 90)
            System.out.println("Letter Grade: A");
        else if (grade >= 80)
            System.out.println("Letter Grade: B");
        else if (grade >= 70)
            System.out.println("Letter Grade: C");
        else if (grade >= 60)
            System.out.println("Letter Grade: D");
        else
            System.out.println("Letter Grade: F");
    }
}
```

Status : Correct

Marks : 10/10

2. Problem Statement

You are given a number of distribution centers (rows) and are tasked with generating a zigzag shipment route pattern. Each shipment route should alternate between left-to-right and right-to-left, as described below.

The program should print the zigzag pattern with a tab (\t) separating the columns. For each row, the shipment numbers should follow a diagonal pattern where numbers are placed in a zigzag, left to right on odd rows and right to left on even rows.

Input Format

The input consists of an integer N, which represents the number of distribution centers (rows) for the zigzag pattern.

Output Format

The output prints the zigzag pattern with N rows, formatted with a tab space (\t) separating the columns.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

Output: 1

```
 2 6
 3 7 10
 4 8 11 13
 5 9 12 14 15
```

Answer

```
// You are using Java
import java.util.Scanner;
public class Main{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
```

```
for(int i=1;i<=n;i++){
    for(int a=i;a<=n;a++){
        System.out.print("");
    }
    int num = i;
    for(int j=1;j<=i;j++){
        System.out.print(num + "\t");
        num+=(n-j);
    }
    System.out.println();
}
input.close();
}
```

Status : Correct

Marks : 10/10

3. Problem Statement

Ravi wants to estimate the total utility bill for a household based on the consumption of electricity, water, and gas.

Write a program to calculate the total bill using the following criteria:

The cost per unit for electricity is 0.12, for water is 0.05, and for gas is 0.08. A discount is applied to the total cost based on the following conditions: If the total cost is 100 or more, a 10% discount is applied. If the total cost is between 50 and 99.99, a 5% discount is applied. No discount is applied if the total cost is less than 50.

The program should output the total bill after applying the discount with two decimal places.

Input Format

The input consists of three double values, representing the number of units consumed for electricity, water, and gas respectively.

Output Format

The output prints a double value, representing the total bill after applying the discount, formatted to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1000.0

200.0

100.0

Output: 124.20

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 electricity = sc.nextDouble();
        double water = sc.nextDouble();
        double gas = sc.nextDouble();

        double cost = (electricity * 0.12) + (water * 0.05) + (gas * 0.08);

        if (cost >= 100)
            cost = cost - (cost * 0.10);
        else if (cost >= 50)
            cost = cost - (cost * 0.05);

        System.out.printf("%.2f", cost);
    }
}
```

Status : Correct

Marks : 10/10

4. Problem Statement

Rohit is tasked with designing a program to analyze the digits of a given integer.

Write a program to help Rohit that takes an integer as input and identifies the minimum odd digit and the maximum even digit present in the number. If no odd or even digits are present, display appropriate messages.

Implement the solution using a 'while' loop to iterate through the digits of the given number.

Input Format

The input consists of an integer n.

Output Format

The first line of output prints the message "Minimum odd digit: " followed by an integer representing the smallest odd digit found in the input number.

If no odd digit exists, it prints "There are no odd digits in the number."

The second line of output prints the message "Maximum even digit: " followed by an integer representing the largest even digit found in the input number.

If no even digit exists, it prints "There are no even digits in the number."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3465

Output: Minimum odd digit: 3

Maximum even digit: 6

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 minOdd = 9;
int maxEven = -1;
boolean hasOdd = false, hasEven = false;

while (n > 0) {
    int digit = n % 10;
    if (digit % 2 == 0) {
        hasEven = true;
        if (digit > maxEven) maxEven = digit;
    } else {
        hasOdd = true;
        if (digit < minOdd) minOdd = digit;
    }
    n /= 10;
}

if (hasOdd)
    System.out.println("Minimum odd digit: " + minOdd);
else
    System.out.println("There are no odd digits in the number.");

if (hasEven)
    System.out.println("Maximum even digit: " + maxEven);
else
    System.out.println("There are no even digits in the number.");
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

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.Scanner;

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

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

Status : Correct

Marks : 10/10

2. Problem Statement

Raj is solving a physics problem involving projectile motion, where he needs to calculate the time a ball hits the ground using a quadratic

equation of the form $ax^2 + bx + c = 0$. Depending on the coefficients, the ball may hit the ground once, twice, or not at all in real time.

Help Raj find all real roots of the equation, if any.

Note: discriminant = $b^2 - 4ac$

Input Format

The input consists of three space-separated doubles a, b, and c, representing the coefficients of the quadratic equation.

Output Format

If there are two real roots, print:

- "Two real solutions:"
- "Root1 = <value>"
- "Root2 = <value>"

If there is one real root, print:

- "One real solution:"
- "Root = <value>"

If there are no real roots, print:

- "There are no real solutions."

Note: values are rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1 6 9

Output: One real solution:

Root = -3.00

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 a = sc.nextDouble();
        double b = sc.nextDouble();
        double c = sc.nextDouble();

        double d = b * b - 4 * a * c;

        if (d > 0) {
            double r1 = (-b + Math.sqrt(d)) / (2 * a);
            double r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.printf("Two real solutions:\nRoot1 = %.2f\nRoot2 = %.2f", r1,
r2);
        } else if (d == 0) {
            double r = -b / (2 * a);
            System.out.printf("One real solution:\nRoot = %.2f", r);
        } else {
            System.out.println("There are no real solutions.");
        }
    }
}
```

Status : Correct

Marks : 10/10

3. Problem Statement

Noah is analyzing numbers within a given range [A, B] and wants to calculate a special sum. For each number in the range, he calculates the product of its odd digits (ignoring even digits). If the number contains no odd digits, it is skipped. The sum of these products for all numbers in the range is the result.

Write a program to compute this sum.

Example

Input:

10 12

Output:

3

Explanation:

For 10, odd digits = 1, product = 1.

For 11, odd digits = 1, 1, product = $1 * 1 = 1$.

For 12, odd digits = 1, product = 1.

Total sum = $1 + 1 + 1 = 3$

Input Format

The input consists of two space-separated integers A and B, representing the inclusive range boundaries.

Output Format

The output prints a single integer representing the sum of the products of odd digits for all numbers in the range.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10 12

Output: 3

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 A = sc.nextInt();
        int B = sc.nextInt();

        int totalSum = 0;
```

```

for (int num = A; num <= B; num++) {
    int temp = num;
    int product = 1;
    boolean hasOdd = false;

    while (temp > 0) {
        int digit = temp % 10;
        if (digit % 2 == 1) {
            product *= digit;
            hasOdd = true;
        }
        temp /= 10;
    }

    if (hasOdd) {
        totalSum += product;
    }
}

System.out.println(totalSum);
}
}

```

Status : Correct

Marks : 10/10

4. Problem Statement

Joe has a favourite number, let's call it X. He wants to check if X is divisible by the sum of its digits. If it is, he considers it a lucky number. If not, he wants to find the closest smaller number, that is divisible by the sum of digits of X. Joe has challenged his friends to solve this puzzle at his birthday party.

Example

Input:

157

Output:

157 is not divisible by the sum of its digits.

The closest smaller number that is divisible: 156

Explanation:

The sum of the digits of X is $1+5+7=13$. Since 157 is not divisible by 13, we need to find the closest smaller number that is divisible by 13. 156 is divisible by 13, it is the closest smaller number that meets the requirement.

Input Format

The input consists of an integer X, representing Joe's favourite number.

Output Format

If X is a lucky number, then the output must be in the format: "X is divisible by the sum of its digits."

If not, then the output must be in the format:

"X is not divisible by the sum of its digits.

The closest smaller number that is divisible: Y",

where X is the entered number and Y is the closest number.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 120

Output: 120 is divisible by 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 X = sc.nextInt();
```

```
int sum = 0, temp = X;
while (temp > 0) {
    sum += temp % 10;
    temp /= 10;
}

if (X % sum == 0) {
    System.out.println(X + " is divisible by the sum of its digits.");
} else {
    System.out.print(X + " is not divisible by the sum of its digits. ");
    int Y = X - 1;
    while (Y > 0 && Y % sum != 0) {
        Y--;
    }
    System.out.println("The closest smaller number that is divisible: " + Y);
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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 secondSmallest = arr[1];  
        int thirdLargest = arr[n - 3];  
  
        System.out.println(secondSmallest + thirdLargest);  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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

```
import java.util.*;  
  
public 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);  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are developing a warehouse management system for a shipping company. The system uses an integer array to represent the weights of packages in a specific order. To verify that the weight capacity is not exceeded, the program needs to calculate the sum of the weights of the first and last packages in the list.

Task:

Write a code to calculate the sum of the weights of the first and last packages in the list. The program should take an integer array as input and return the total weight of the first and last packages.

Input Format

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

The second line of the input is N space-separated integer values.

Output Format

The output is displayed in the following format:

"Sum of the first and last elements: <>Sum<>"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

10 20 30 40 50

Output: Sum of the first and last elements: 60

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();  
        }  
  
        int sum = arr[0] + arr[n - 1];  
  
        System.out.println("Sum of the first and last elements: " + sum);  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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

```
import java.util.*;

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

        // Input number of rows and 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];

        // Input first matrix
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
```

```
        matrix1[i][j] = sc.nextInt();
    }

// Input second matrix
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        matrix2[i][j] = sc.nextInt();
    }
}

// Add 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 result
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        System.out.print(result[i][j] + " ");
    }
    System.out.println();
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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

```
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();  
        }  
  
        Set<Integer> seen = new HashSet<>();  
        int firstRepeated = -1;  
  
        for (int i = 0; i < n; i++) {  
            if (seen.contains(arr[i])) {  
                firstRepeated = arr[i];  
                break;  
            }  
            seen.add(arr[i]);  
        }  
    }  
}
```

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

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 3_PAH

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Eminem is a billiard player who enjoys playing billiards and also likes solving mathematical puzzles. He notices that the billiard balls on the table are arranged in a grid, and he is curious to find the sum of the numbers written on each ball.

Write a program to find the sum of all the numbers written on each ball in the grid.

Input Format

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

The second line consists of an integer M, representing the number of columns.

The following lines N lines consist of M space-separated integers, representing the numbers written on each ball.

Output Format

The output prints an integer representing the sum of all the numbers written on each ball.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3

3

1 2 3

4 5 6

7 8 9

Output: 45

Answer

```
import java.util.*;

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

        int n = sc.nextInt();
        int m = sc.nextInt();

        int sum = 0;

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                sum += sc.nextInt();
            }
        }

        System.out.println(sum);
    }
}
```

2. Problem Statement

Egath is participating in a coding hackathon, and one of the challenges requires him to work with an array of integers. The task is to remove exactly one element from the array such that the sum of the remaining elements is a prime number.

Help Egath find the first possible prime sum by removing one element or determining if no such prime sum can be achieved.

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 removing one element results in a prime sum, print the sum.

If no such prime sum can be achieved by removing exactly one element, print "No valid prime sum found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3

1 2 3

Output: 5

Answer

```
import java.util.*;  
public class Main {
```

```

public static boolean isPrime(int num) {
    if (num < 2) return false;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) return false;
    }
    return true;
}

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

    int n = sc.nextInt();
    int[] arr = new int[n];

    int totalSum = 0;
    for (int i = 0; i < n; i++) {
        arr[i] = sc.nextInt();
        totalSum += arr[i];
    }

    boolean found = false;

    for (int i = 0; i < n; i++) {
        int newSum = totalSum - arr[i];
        if (isPrime(newSum)) {
            System.out.println(newSum);
            found = true;
            break;
        }
    }

    if (!found) {
        System.out.println("No valid prime sum found");
    }
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

In a customer loyalty program, reward points are logged in a sorted array as customers make transactions. Occasionally, due to system errors, duplicate entries for the same transaction may appear. To ensure accurate reward calculations, it's crucial to remove these duplicates from the list.

Write a program to process the array of reward points, removing any duplicates while preserving the order of unique entries. The program should then display the cleaned list of unique reward points and the total count of these unique points.

Input Format

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

The second line consists of N space-separated integers, representing the reward points in sorted order.

Output Format

The first line of output prints the cleaned list of unique reward points separated by a space.

The second line of output prints an integer representing the total count of unique reward points.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
100 100 200

Output: 100 200
2

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();
}

LinkedHashSet<Integer> unique = new LinkedHashSet<>();
for (int num : arr) {
    unique.add(num);
}
for (int num : unique) {
    System.out.print(num + " ");
}
System.out.println();

System.out.println(unique.size());
}
```

Status : Correct

Marks : 10/10

4. Problem Statement

Priya is building a system to automate image transformations using matrix operations. To do this, she needs to multiply two matrices representing pixel data and transformation rules.

Help Priya perform matrix multiplication and print the resulting matrix if the operation is valid.

Input Format

The first line of input consists of two int values, representing the number of rows R1 and columns C1 of the first matrix.

The next $R1 \times C1$ integers represent the elements of the first matrix.

The next line consists of two int values, representing the number of rows R2 and columns C2 of the second matrix.

The next $R_2 \times C_2$ integers represent the elements of the second matrix.

Output Format

If matrix multiplication is possible, print R_1 lines, each containing C_2 space-separated int values representing the resulting matrix.

Otherwise, print "Matrix multiplication not possible".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2 3

1 2 3

4 5 6

3 2

7 8

9 10

11 12

Output: 58 64

139 154

Answer

```
import java.util.*;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        int R1 = sc.nextInt();  
        int C1 = sc.nextInt();  
        int[][] A = new int[R1][C1];  
  
        for (int i = 0; i < R1; i++) {  
            for (int j = 0; j < C1; j++) {  
                A[i][j] = sc.nextInt();  
            }  
        }  
    }  
}
```

```
        }

        int R2 = sc.nextInt();
        int C2 = sc.nextInt();
        int[][] B = new int[R2][C2];

        for (int i = 0; i < R2; i++) {
            for (int j = 0; j < C2; j++) {
                B[i][j] = sc.nextInt();
            }
        }

        if (C1 != R2) {
            System.out.println("Matrix multiplication not possible");
            return;
        }

        int[][] result = new int[R1][C2];

        for (int i = 0; i < R1; i++) {
            for (int j = 0; j < C2; j++) {
                for (int k = 0; k < C1; k++) {
                    result[i][j] += A[i][k] * B[k][j];
                }
            }
        }

        for (int i = 0; i < R1; i++) {
            for (int j = 0; j < C2; j++) {
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Ubendra L
Email: 241901120@rajalakshmi.edu.in
Roll no: 241901120
Phone: 8610285702
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 3_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Rina is managing the inventory for a library, where each row of a 2D matrix represents the number of different genres of books available on each shelf.

She wants to perform the following operations:

Transformation: Replace each element in a row with the sum of all elements in that row.
Merging: After transformation, Rina will provide one additional matrix, and specify whether to merge the transformed matrix with this new matrix row-wise or column-wise.

Input Format

The first line contains two integers R and C, representing the number of rows and columns of the initial matrix.

The next R lines contain C space-separated integers, representing the book counts in the library.

The next line contains two integers MR and MC, representing the dimensions of the second matrix (to be merged).

The next MR lines contain MC space-separated integers, representing the second matrix.

The last line contains an integer mergeType:

- 0 Row-wise merging (append the second matrix below the transformed matrix).
- 1 Column-wise merging (append the second matrix to the right of the transformed matrix).

Output Format

The output prints "Transformed matrix: " followed by the transformed 2D matrix where each element in a row is replaced with the sum of the elements in that row.

The output prints "Final merged matrix: ", followed by the merging based on mergeType.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3 4
8 2 4 9
4 5 6 1
7 8 9 3
2 4
3 5 7 2
6 1 4 9
0

Output: Transformed matrix:

23 23 23 23
16 16 16 16

27 27 27 27

Final merged matrix:

23 23 23 23

16 16 16 16

27 27 27 27

3 5 7 2

6 1 4 9

Answer

```
import java.util.Scanner;

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

        int R = scanner.nextInt();
        int C = scanner.nextInt();
        int[][] matrix1 = new int[R][C];

        for (int i = 0; i < R; i++) {
            for (int j = 0; j < C; j++) {
                matrix1[i][j] = scanner.nextInt();
            }
        }

        int MR = scanner.nextInt();
        int MC = scanner.nextInt();
        int[][] matrix2 = new int[MR][MC];

        for (int i = 0; i < MR; i++) {
            for (int j = 0; j < MC; j++) {
                matrix2[i][j] = scanner.nextInt();
            }
        }

        int mergeType = scanner.nextInt();

        int[][] transformedMatrix = new int[R][C];
        for (int i = 0; i < R; i++) {
            int sum = 0;
            for (int j = 0; j < C; j++) {
                sum += matrix1[i][j];
            }
            transformedMatrix[i][0] = sum;
        }
    }
}
```

```
        }
        for (int j = 0; j < C; j++) {
            transformedMatrix[i][j] = sum;
        }
    }

System.out.println("Transformed matrix:");
for (int i = 0; i < R; i++) {
    for (int j = 0; j < C; j++) {
        System.out.print(transformedMatrix[i][j]);
        if (j < C - 1) {
            System.out.print(" ");
        }
    }
    System.out.println();
}

int[][] mergedMatrix;
if (mergeType == 0) {
    mergedMatrix = new int[R + MR][C];
    for (int i = 0; i < R; i++) {
        System.arraycopy(transformedMatrix[i], 0, mergedMatrix[i], 0, C);
    }
    for (int i = 0; i < MR; i++) {
        System.arraycopy(matrix2[i], 0, mergedMatrix[i + R], 0, MC);
    }
} else {
    mergedMatrix = new int[R][C + MC];
    for (int i = 0; i < R; i++) {
        System.arraycopy(transformedMatrix[i], 0, mergedMatrix[i], 0, C);
        System.arraycopy(matrix2[i], 0, mergedMatrix[i], C, MC);
    }
}

System.out.println("Final merged matrix:");
int rows = mergedMatrix.length;
int cols = mergedMatrix[0].length;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        System.out.print(mergedMatrix[i][j]);
        if (j < cols - 1) {
            System.out.print(" ");
        }
    }
}
```

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

scanner.close();
}
}
```

Status : Correct

Marks : 10/10

2. Problem Statement

Emma is a data analyst working with a grid-based system where each cell contains important numerical data. The grid represents spatial data, inventory records, or structured reports that require periodic updates.

Due to system updates and new requirements, Emma needs to modify the grid in the following ways:

She wants to insert either a new row or a new column at a given position. Later, she needs to delete either a row or a column from the modified matrix.

Input Format

The first line contains two integers rows and cols (the dimensions of the matrix).

The next rows lines contain cols space-separated integers representing the initial matrix.

The next line contains two integers insertType and insertIndex:

- insertType = 0 for row insertion, 1 for column insertion.
- insertIndex is the position where the new row/column should be added.

If inserting a row, the next cols integers represent the new row or If inserting a column, the next rows integers represent the new column.

The next line contains two integers deleteType and deleteIndex:

- deleteType = 0 for row deletion, 1 for column deletion.
- deleteIndex is the position to be deleted.

Output Format

The first line of output prints the string "After insertion" followed by the modified matrix with the inserted row or column.

Each row of the matrix is printed on a new line with space-separated integers.

The next line prints the string "After deletion" followed by the final matrix after the specified deletion operation.

Each row of the resulting matrix is printed on a new line with space-separated integers.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3 3

1 2 3

4 5 6

7 8 9

0 1

10 11 12

1 2

Output: After insertion

1 2 3

10 11 12

4 5 6

7 8 9

After deletion

1 2

10 11

4 5

7 8

Answer

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

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        int rows = scanner.nextInt();  
        int cols = scanner.nextInt();  
        int[][] matrix = new int[rows][cols];  
  
        for (int i = 0; i < rows; i++) {  
            for (int j = 0; j < cols; j++) {  
                matrix[i][j] = scanner.nextInt();  
            }  
        }  
  
        int insertType = scanner.nextInt();  
        int insertIndex = scanner.nextInt();  
        int[] newRow = null;  
        int[] newCol = null;  
  
        if (insertType == 0) {  
            newRow = new int[cols];  
            for (int j = 0; j < cols; j++) {  
                newRow[j] = scanner.nextInt();  
            }  
        } else {  
            newCol = new int[rows];  
            for (int i = 0; i < rows; i++) {  
                newCol[i] = scanner.nextInt();  
            }  
        }  
  
        int deleteType = scanner.nextInt();  
        int deleteIndex = scanner.nextInt();  
  
        // Perform insertion  
        int[][] afterInsertion;  
        if (insertType == 0) {  
            afterInsertion = new int[rows + 1][cols];  
            for (int i = 0; i < insertIndex; i++) {  
                System.arraycopy(matrix[i], 0, afterInsertion[i], 0, cols);  
            }  
            System.arraycopy(newRow, 0, afterInsertion[insertIndex], 0, cols);  
        }  
    }  
}
```

```
        for (int i = insertIndex; i < rows; i++) {
            System.arraycopy(matrix[i], 0, afterInsertion[i + 1], 0, cols);
        }
    } else {
        afterInsertion = new int[rows][cols + 1];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < insertIndex; j++) {
                afterInsertion[i][j] = matrix[i][j];
            }
            afterInsertion[i][insertIndex] = newCol[i];
            for (int j = insertIndex; j < cols; j++) {
                afterInsertion[i][j + 1] = matrix[i][j];
            }
        }
    }

System.out.println("After insertion");
for (int i = 0; i < afterInsertion.length; i++) {
    for (int j = 0; j < afterInsertion[i].length; j++) {
        System.out.print(afterInsertion[i][j]);
        if (j < afterInsertion[i].length - 1) {
            System.out.print(" ");
        }
    }
    System.out.println();
}

// Perform deletion
int[][] afterDeletion;
if (deleteType == 0) {
    int newRows = afterInsertion.length - 1;
    int newCols = afterInsertion[0].length;
    afterDeletion = new int[newRows][newCols];
    for (int i = 0; i < deleteIndex; i++) {
        System.arraycopy(afterInsertion[i], 0, afterDeletion[i], 0, newCols);
    }
    for (int i = deleteIndex + 1; i < afterInsertion.length; i++) {
        System.arraycopy(afterInsertion[i], 0, afterDeletion[i - 1], 0, newCols);
    }
} else {
    int newRows = afterInsertion.length;
    int newCols = afterInsertion[0].length - 1;
```

```

        afterDeletion = new int[newRows][newCols];
        for (int i = 0; i < newRows; i++) {
            for (int j = 0; j < deleteIndex; j++) {
                afterDeletion[i][j] = afterInsertion[i][j];
            }
            for (int j = deleteIndex + 1; j < afterInsertion[i].length; j++) {
                afterDeletion[i][j - 1] = afterInsertion[i][j];
            }
        }
    }

    System.out.println("After deletion");
    for (int i = 0; i < afterDeletion.length; i++) {
        for (int j = 0; j < afterDeletion[i].length; j++) {
            System.out.print(afterDeletion[i][j]);
            if (j < afterDeletion[i].length - 1) {
                System.out.print(" ");
            }
        }
        System.out.println();
    }

    scanner.close();
}
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

Nikila is working as an intern in a software firm and is practicing with a matrix where each row represents a set of numerical values. Her task is to identify the row with the highest sum of its elements and remove that row from the matrix. After removing the row with the highest sum, Nikila needs to print the updated matrix.

Your task is to help Nikila in implementing the same. If there are two or more rows that have same the highest sum, the firstly encountered row is deleted.

Input Format

The first line of the input consists of two space-separated integers, R and C, representing the number of rows and columns in the matrix, respectively.

The following R lines each contain, C space-separated integers representing the matrix elements.

Output Format

The output prints the matrix after removing the row with the highest sum. Each row should be printed on a new line, with elements separated by a space.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 2 2

1 2

3 4

Output: 1 2

Answer

```
import java.util.Scanner;

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

        int R = scanner.nextInt();
        int C = scanner.nextInt();
        int[][] matrix = new int[R][C];

        for (int i = 0; i < R; i++) {
            for (int j = 0; j < C; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }

        int maxSum = Integer.MIN_VALUE;
        int rowToRemove = -1;
```

```

for (int i = 0; i < R; i++) {
    int sum = 0;
    for (int j = 0; j < C; j++) {
        sum += matrix[i][j];
    }
    if (sum > maxSum) {
        maxSum = sum;
        rowToRemove = i;
    }
}

int[][] newMatrix = new int[R-1][C];
int newIndex = 0;
for (int i = 0; i < R; i++) {
    if (i != rowToRemove) {
        System.arraycopy(matrix[i], 0, newMatrix[newIndex], 0, C);
        newIndex++;
    }
}

for (int i = 0; i < newMatrix.length; i++) {
    for (int j = 0; j < newMatrix[i].length; j++) {
        System.out.print(newMatrix[i][j]);
        if (j < newMatrix[i].length - 1) {
            System.out.print(" ");
        }
    }
    System.out.println();
}

scanner.close();
}
}

```

Status : Correct

Marks : 10/10

4. Problem Statement:

Imagine you have an array of integer values, and you're tasked with identifying a pair of elements within the array. This pair of elements should

have a sum that is the closest to zero when compared to any other pair in the array.

Your goal is to create a program that solves this problem efficiently. The program should accept an array of integers and return the pair of elements whose sum is closest to zero.

Input Format

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

The second line of the input contains N space-separated integer values.

Output Format

The output is displayed in the following format:

"Pair with the sum closest to zero: {value} and {value}"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

9 10 -3 -5 -2

Output: Pair with the sum closest to zero: 9 and -5

Answer

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

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

        int n = scanner.nextInt();
        int[] arr = new int[n];

        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
    }
}
```

```
int minAbsSum = Integer.MAX_VALUE;
int firstElement = 0;
int secondElement = 0;

for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {
        int sum = arr[i] + arr[j];
        int absSum = Math.abs(sum);
        if (absSum < minAbsSum) {
            minAbsSum = absSum;
            firstElement = arr[i];
            secondElement = arr[j];
        }
    }
}

System.out.println("Pair with the sum closest to zero: " + firstElement +
and " + " + secondElement);

scanner.close();
}
```

Status : Correct

Marks : 10/10