**Question no 1)Problem Statement**

Dharun is a teacher and has just finished grading the exams of his students. He wants to analyze the performance of his students by finding the smallest difference between the scores of any two students. Help him write the formula that will help him calculate this smallest difference so that he can generate the report for the same using pointer arithmetic.

**Function Name:** int findSmallestDifference - This function calculates and returns the smallest difference between any two students' scores.

**Example**

**Input:**

5

100 89 95 56 66

**Output:**

5

**Explanation:**

Here the smallest difference between any two scores (100 and 95), which in this case is 5. This indicates that the closest scores among the students differ by a margin of 5 points.

**Company Tags:** TCS

**Input format :**

The first line of input consists of the number of students, **n**as an integer.

The second line of input consists of **n** space-separated integers representing the scores.

**Output format :**

The output displays the smallest difference between any two scores.

**Refer to the sample output for formatting specifications.**

**Code constraints :**

In this scenario, the test cases fall under the following constraints:

2 ≤ n ≤ 10

0 ≤ scores ≤ 100

**Sample test cases :**

**Input 1 :**

5

100 89 95 56 66

**Output 1 :**

5

**Input 2 :**

3

99 100 90

**Output 2 :**

1

**Question-2)Problem Statement**

Dhruv is a professor who wants to develop a program to manage student information using a class called **Student**. The program should allow Dhruv to input details about multiple students, such as their names, the number of subjects they have, and their marks in each subject.

To calculate and display the average marks for each student, Dhruv intends to use **pointers to objects** in his program. Help Dhruv to write a program.

**Input format :**

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

For each student, the following details should be entered:

1. The first line of input consists of a string, representing the name of the student.
2. The second line of input consists of an integer**m,** representing the number of subjects for that student.
3. The last line of input consists of **m** double values, representing the marks obtained in each subject separated by space.

**Output format :**

For n students, the program should display the following for each student:

1. The first line of output displays the name of the student.
2. The next line of output displays the average marks obtained by the student as a double value rounded off to two decimal places.

**Refer to the sample output for formatting specifications.**

**Code constraints :**

In this scenario, the test cases fall under the following constraints:

1 ≤ n ≤ 100

1 ≤ m ≤ 10

1.0 ≤ marks ≤ 100.0

**Sample test cases :**

**Input 1 :**

3

John Doe

5

80.5 90.0 75.5 85.0 92.0

Jane Smith

4

88.0 92.5 76.0 85.5

Alice Johnson

3

70.0 65.5 80.0

**Output 1 :**

Student Details:

Name: John Doe

Average marks: 84.60

Name: Jane Smith

Average marks: 85.50

Name: Alice Johnson

Average marks: 71.83

**Input 2 :**

2

Pinky

3

98.0 76.0 43.0

Rosy

4

98.0 61.0 54.0 37.0

**Output 2 :**

Student Details:

Name: Pinky

Average marks: 72.33

Name: Rosy

Average marks: 62.50

**Question-3)Problem Statement**

Janu enjoys learning puzzles from her childhood. She is given an **rxc** matrix and asked to sum the elements in each row. She needs to attain the sum of elements in a matrix. Create a class named **MatrixSum** to handle the same.

**Input format :**

The first line contains an integer **r,** representing the number of rows in the matrix.

The second line contains an integer **c,** representing the number of columns in the matrix.

Following these, there are r lines, each containing c integers. These integers represent the elements of the matrix, row by row.

**Output format :**

The output prints elements of each row along with the sum of each row.

**Refer to the output for formatting specifications.**

**Code constraints :**

In this scenario, the test cases fall under the following constraints:

1 ≤ r, c ≤ 10

0 ≤ elements ≤ 1000

**Sample test cases :**

**Input 1 :**

2

3

27 38 59

73 88 99

**Output 1 :**

27 38 59 SUM: 124

73 88 99 SUM: 260

**Input 2 :**

3

4

1 2 3 4

5 6 7 8

9 3 2 4

**Output 2 :**

1 2 3 4 SUM: 10

5 6 7 8 SUM: 26

9 3 2 4 SUM: 18

**Question-4)Problem Statement**

John and Michael play a game of multiplication. The game starts with an integer p initialized to 1. They take turns multiplying p by one of the numbers from 1 to 9 in a sequence. John always starts by multiplying p by 1 and then passes the result to Michael. Michael multiplies the number by 2 and passes the result back to John. John then multiplies by 3, and so on, increasing the multiplier by 1 with each turn.

Before the game starts, they draw an integer N. The winner is the player who first reaches or exceeds p >= N during their turn. Write a class **MultiplicationGame** with a member function **void game(int)** to simulate this scenario and determine the winner.

**Example**

**Input:** N=10

**Output:**

10 Michael wins

**Explanation**

If N is 10:

John starts with p = 1, multiplies by 1 (John's turn), resulting in p = 1.

Michael multiplies p = 1 by 2, resulting in p = 2.

John multiplies p = 2 by 3, resulting in p = 6.

Michael multiplies p = 6 by 4, resulting in p = 24.

Since 24 is greater than 10, Michael wins.

**Company Tags:** Accenture

**Input format :**

The input consists of an integer **N.**

**Output format :**

The output prints the value of N and the name of the winner, separated by a space.

**Refer to the sample outputs for the formatting specifications.**

**Code constraints :**

In this scenario, the test cases fall under the following constraints:

1 ≤ N ≤ 20

**Sample test cases :**

**Input 1 :**

10

**Output 1 :**

10 Michael wins

**Input 2 :**

3

**Output 2 :**

3 John wins

**Question-5)Problem Statement**

Write a function **isValidPhoneNumber** that takes a phone number (string) as input and determines whether it is a valid phone number. The phone number format should be XXX-XXX-XXXX, where X represents a digit. The number is considered valid if the hyphens are in exact same positions and the length of the string is 12.

**Input format :**

The input consists of a phone number in the format XXX-XXX-XXXX, where X represents a digit.

**Output format :**

The output displays whether the entered string is a valid phone number or not.

If the number is valid, the output displays the number without hyphens.

**Refer to the sample output for formatting specifications.**

**Code constraints :**

The phone number should be in the format XXX-XXX-XXXX, with a total length of 12 characters.

Each X in the phone number should be a digit (0-9).

No additional characters or spaces should be present in the input.

**Sample test cases :**

**Input 1 :**

123-456-7890

**Output 1 :**

Yes, the phone number is valid

1234567890

**Input 2 :**

123-456-789

**Output 2 :**

No, the phone number is not valid

**Input 3 :**

123456-7890

**Output 3 :**

No, the phone number is not va