Task #1

- Using pointers to process arrays (50 points)

An arithmetic sequence is a sequence of numbers such that the difference from any succeeding term to its preceding term remains constant throughout the sequence. For instance, the sequence {4, 7, 10, 13} with a constant difference equal to 3, and the sequence {5, 15, 20, 25} is an arithmetic sequence with constant difference equal to 5. The sequence {3, 5, 6} is not an arithmetic sequence, since the distance between the 1st and 2nd elements is 2, but the distance between the 2nd and 3rd elements is 1.

Write a program that determines if a sequence is an arithmetic sequence that includes the function is_arithmetic_sequence that checks if an input sequence is an arithmetic sequence.

Use pointers to process the array in the assign function.

Requirements

- The program should include the following function:
  void assign(int *points, int *team_assignment, int n);
  he function calculates the team assignment and store the results in the team_assignment array . Array points represents the points for each students. n is the total number of students.

This function should use pointer arithmetic– not subscripting – to visit array elements. In other words, eliminate the loop index variables and all use of the [] operator in the function.

- In the main function, call the **assign** function and display the result.
- Pointer arithmetic is NOT required in the main function.

Examples (your program must follow this format precisely)

Example #1

```
Enter number of students: 4

Enter points for each students: 72 25 128 65

Beginner team: student 2

Honor team: student 1 4

Excellence team: student 3
```

Example #2

```
Enter number of students: 7

Enter points for each students: 89 34 143 74 162 23 65

Beginner team: student 2 6

Honor team: student 1 4 7

Excellence team: student 3 5
```

Task #2

- - Using pointers to process arrays (50 points)

An arithmetic sequence is a sequence of numbers such that the difference from any succeeding term to its preceding term remains constant throughout the sequence. For instance, the sequence **{4, 7, 10, 13}** with a constant difference equal to **3**, and the sequence **{5, 15, 20, 25}** is an arithmetic sequence with constant difference equal to **5**. The sequence **{3, 5, 6}** is not an arithmetic sequence, since the distance between the 1st and 2nd elements is **2**, but the distance between the 2nd and 3rd elements is **1**.

Write a program that determines if a sequence is an arithmetic sequence that includes the function **is_arithmetic_sequence** that checks if an input sequence is an arithmetic sequence.

Requirements

- The program should include the following function:
  int is_arithmetic_sequence(int *sequence, int n);
  This function returns 1 if sequence is an arithmetic sequence, and returns 0 otherwise. n is the size of the sequence (array).

The function should use pointer arithmetic – not subscripting – to visit array elements. In other words, eliminate the loop index variables and all use of the [] operator in the function.

- Follow the format of the examples below.

- In the main function, ask the user to enter the size of the sequence and the numbers of the sequence.
- **The main function call the is_arithmetic_sequence function and display the result.**
- Pointer arithmetic is NOT required in the main function.

Examples (your program must follow this format precisely)

Example #1

```
Enter length of the sequence: 5

Enter numbers of the sequence: 3 17 32 56 88

no
```

Example #2

```
Enter length of the sequence: 5

Enter numbers of the sequence: 3 10 17 24 31

yes
```

Example #3

```
Enter length of the sequence: 7

Enter numbers of the sequence: 13 17 21 25 29 33 37

yes
```