TP8-Using array (part 1)

Lab's objectives:

- Using to store multiple data of the same type in one special variable

Deadline: 6 days

Problem1: Write a program in C to ask a user for 10 numbers and store them in an array Display all numbers in array, named ***numbers***.

Example:

INPUT:

Enter number 1: 20
Enter number 2: -2
Enter number 3: 9
Enter number 4: 77
Enter number 5: 0
Enter number 6: 12
Enter number 7: 10
Enter number 8: 8
Enter number 9: 6
Enter number 10: 2

Output:

All data in the array are: 20, -2, 9, 77, 0, 12, 10, 8, 6, 2.

Problem2: Similarly to exercise in problem1, we also compute some statistical data such as the total summation, average, and find the max number. Assume that the size of this array is 7.

Example:

INPUT:

Enter number 1: 12
Enter number 2: 4
Enter number 3: -6
Enter number 4: 80
Enter number 5: 0
Enter number 6: -3
Enter number 7: 7

Output:

All data in the array are: 12, 4, -6, 80, 0, -3, 7
Computed statistical data are:
Total= 94,
Average is 13.49, and
The max number is 80.

Problem3: Write a program in C to store 5 numbers in an array called **a1**.
Then create another array called **a2**. Get all data in **a1** and store in **a2** so that these two arrays contain the same data. Display all numbers in a2 from the start position to the end. Also, display all numbers in a2 from the end position back to the start position.

Example:

INPUT:

***** Getting data for the array a1**

Enter number 1: 2
Enter number 2: 4
Enter number 3: 6
Enter number 4: 20
Enter number 5: 125

Output:

All data input by the users are stored successfully in the array **a1**.
A new array a2 has been created and it has the same data as a1.
Display a2 from start to end position: 2 4 6 20 125
Display a2 from end to start position: 125 20 6 4 2

Problem4: Write a C program to store data of 4 students. Each student has a name, score, email and ages. The program asks users to put data (let input your friends' information). Display all students' information on screen.
Hint: Create 4 arrays namely **names**, **scores**, **emails**, **ages.** Each array has the size of 4.

```
char names[4][100];
float scores[4];
char emails[4][100];
int ages[4];
```

Example:

INPUT:

***** Getting data for student 1:**

Name: Dara
Age: 20
Email: dara@gamil.com
Score: 90

***** Getting data for student 2:**

Name: Pisey
Age: 22
Email: pisey@gamil.com
Score: 84

*** **Getting data for student 3:**
Name: Sok
Age: 21
Email: sok@gamil.com
Score: 95

*** **Getting data for student 4:**
Name: Seyha
Age: 19
Email: seyha@gamil.com
Score: 96

Output:

All data of students in the array are:

Name	**Age**	**Score**	**Email**
Dara	20	90	dara@gmailcom
Pisey	22	84	pisey@gmailcom
Sok	21	95	sok@gmailcom
Seyha	19	96	seyha@gmailcom

Problem5: Similar to problem 4. Find an average score and display info of all students who got scores more than the average. The size of the array to store students' data is 5.

Example:

INPUT:

*** **Instruction:** Enter name, age, score and email separated by space:

Input for student 1: Dara 20 58 dara@gmail.com
Input for student 2: Sok 22 45 dara@gmail.com
Input for student 3: Sao 18 90 dara@gmail.com
Input for student 4: Jack 19 80 dara@gmail.com
Input for student 5: Rose 21 55 dara@gmail.com

OUTPUT:

The average score is: 65.6
All students got score more than the average are:

Name	**Age**	**Score**	**Email**
Sao	18	90	sao@gmailcom
Jack	19	80	jack@gmailcom