MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG140 C Programming – Worksheet 16

Purpose:
In this work sheet, you will use practice pointers, pointer arithmetic, pointers and arrays, and will do some exercises on dynamically created arrays.

Tasks:

a. Write a program that dynamically creates two arrays called a and b, and then generates their inner product. The program needs to get the array size from the user which is of length n. The inner product of two arrays are calculated as follows:

a[0]*b[0]+a[1]*b[1]+a[2]*b[2]+…+a[n-1]b[n-1]

Use pointer arithmetic-not subscripting-to visit array elements.

A sample run would be as follows:

```
How many elements you want to store (n)? 3
Enter a value to be stored in the first array: 2
Enter a value to be stored in the first array: 3
Enter a value to be stored in the first array: 2

Enter a value to be stored in the second array: 3
Enter a value to be stored in the second array: 2
Enter a value to be stored in the second array: 3

Result of inner product is: 18
```

b. Consider a sequence of floating-point numbers, x_i, I = 1,2,3,…m.

The mean is defined as

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_m}{m}$$

The deviation about the mean is

$$d_i = (x_i - \bar{x}) \qquad i = 1,2,\ldots,m$$

And the standard deviation is

$$s = \sqrt{({d_1}^2 + {d_2}^2 + \cdots + {d_m}^2)/m}$$

Write a program that ask user how many numbers they want to use in their statistic called m and reads in m elements into one-dimensional floating point array, and then calculates the sum of these elements, the mean, the standard deviation.

A sample run would be as follows:

```
How many numbers you want to use in your statistics? 10
Please enter 10 numbers: 27.5 13.4 53.8 29.2 74.5 87.0 39.9 47.7 8.1
63.2
```

```
Sum: 444.3
Mean: 44.4
Standard deviation: 24.47 (Note: this is an approximation and could
be sligthly different on your machine)
```

c. Consider the following foreign currencies and their equivalents to one Turkish Lira:

US Dollar: 6.65 per Turkish Lira
Australian Dollar: 4.04 per Turkish Lira
Euro: 7.26 per Turkish Lira
British Pound: 8.22 per Turkish Lira
Indian Ruppe: 0 09 per Turkish Lira
Canadian Dollar: 4.68 per Turkish Lira
Bahraini Dinar: 17.69 per Turkish Lira
Cuban Peso: 0.25 per Turkish Lira
Japanese Yen: 0.06 per Turkish Lira
French Franc: 1.11 per Turkish Lira
Mexican Peso: 0.28 per Turkish Lira
Brazilian Real: 1.28 per Turkish Lira

Write an interactive menu-driven program that will accept two different currencies and return the value of the second currency per one unit of the first currency. For example, if the two currencies are Japanese yen and Mexican pesos, the program will return the number of Mexican pesos equivalent to one Japanese Yen. Use the data given above to carry out the conversions. Design your program such that it continous repetitiously until an ending condition is selected from the menu. You need to use script notation for your array.

A sample run would be as follows:

```
**********************************************************************
Currency Exchange Menu

Select (c) for conversion
Select (e) for exit

Enter your choice: c
**********************************************************************
Currency List
(1)US Dollar
(2)Australian Dollar
(3)Euro
(4)British Pound
(5)Indian Ruppe
(6)Canadian Dollar
(7)Bahraini Dinar
(8)Cuban Peso
(9)Japanese Yen
(10)French Franc
(11)Mexican Peso
(12)Brazilian Real

Enter your choice for currency from: 1
Enter your choice for currency to: 5
```

```
Enter the amount: 25

25 US Dollar is 1847.22 Indian Ruppe

*********************************************************************
Currency Exchange Menu

Select (c) for conversion
Select (e) for exit

Enter your choice: c
*********************************************************************
Currency List
(1)US Dollar
(2)Australian Dollar
(3)Euro
(4)British Pound
(5)Indian Ruppe
(6)Canadian Dollar
(7)Bahraini Dinar
(8)Cuban Peso
(9)Japanese Yen
(10)French Franc
(11)Mexican Peso
(12)Brazilian Real

Enter your choice for currecny from: 12
Enter your choice for currency to: 8
Enter amount: 300

300 Brazilian Real is 1536 Cuban Peso

*********************************************************************
Currency Exchange Menu

Select (c) for conversion
Select (e) for exit

Enter your choice: e
*********************************************************************
```

d. Body Mass Index (BMI) is a person's weight in kilograms divided by the square of height in meters. A high BMI can indicate high body fatness, and a low BMI can indicate too low body fatness. For axample, BMA calcultaion is as follows;

Weight = 68 kg, Height = 165 cm, then BMI= 68 / $(1.65)^2$ = 24.98

You need to write a program which creates a BMI table (a two dimensional array). Array will contain BMI values for height from 145 to 190 cm (incrementing by 5) and for weight from 45 to 90 (incrementing by 5).

Then, your program will fill the array as follows;

- If BMI is 18.4 or less, it falls within the underweight range, fill array with -1.
- If BMI is 18.5 to 24.9, it falls within the normal or healthy weight range, fill array with 0.
- If BMI is 25.0 to 29.9, it falls within the overweight range, fill array with 1.
- If BMI is 30.0 or higher, it falls within the obese range, fill array with 2.

A sample run would be as follows:

```
BMI Table for height 145-190 cm and weight 45-90 kg:
(-1): Underweight range
(0) : Healthy Weight range
(1) : Overweight range
(2) : Obese range
```

	45	50	55	60	65	70	75	80	85	90
145	0	0	1	1	2	2	2	2	2	2
150	0	0	0	1	1	2	2	2	2	2
155	0	0	0	1	1	1	2	2	2	2
160	-1	0	0	0	1	1	1	2	2	2
165	-1	-1	0	0	0	1	1	1	2	2
170	-1	-1	0	0	0	0	1	1	1	2
175	-1	-1	-1	-1	0	0	0	1	1	1
180	-1	-1	-1	-1	0	0	0	0	1	1
185	-1	-1	-1	-1	0	0	0	0	0	1
190	-1	-1	-1	-1	-1	0	0	0	0	0

e. Imagine that you have two randomly populated arrays and asked to write a program to find out which one is more similar to the array given by the user. In your program you will first ask user to enter the size of arrays (assume N). Then, you will create two arrays dynamically with the given size (N). For example, if the size is 10 (N=10), you will create two arrays dynamically with the size of 10. Then, populate these arrays with randomly generated numbers. Then, you will dynamically create another array with the given size and ask user to enter its elements. Finally, you will measure the similarity between the given array and the randomly generated two arrays using the Euclidean distance measure given below. As a result, the given array will be more similar to the one which has the minimum distance (higher similarity).

$$Distance(x, y) = \sqrt{\sum_{i=0}^{N-1} (x_i - y_i)^2}$$

Please note that in this formula, x and y are represening two arrays and x_i is rerpresenting the element in array x at the position of i.

A sample run would be as follows:

```
Enter the array size: 10

First array created as: 80 70 41 68 96 52 39 36 98 89
Second array created as: 3 7 23 10 5 13 12 7 9 20

Enter your array data: 5 10 20 6 1 8 15 11 13 14

Distance of your array to the first array: 196.2702
Distance of your array to the second array: 12.4900

Your array is more similar to the second array!
```

Recommended Reading: Chapter 11 (p. 539-553)
Recommended Exercises: Programming Exercises 11.1 and 11.2.