East West University
Department of Computer Science & Engineering
Plot No-A/2, Main Road, Jahurul Islam City, Aftabnagar, Dhaka-1219

Lab Manual
Course : CSE -103
Credit Title : Structured Programming
Instructor : Dr. Maheen Islam, Associate Professor, CSE Dept, East West University

Lab - 5: Array

Exercise 1:
Type the following code and match your output with the given output.

```
#include <stdio.h>
main()
{
  float expenses[13];
  int count;
   for (count = 1; count < 13; count++)
   {
      printf("Enter expenses for month %d: ", count);
      scanf("%f", &expenses[count]);
   }
   for (count = 1; count < 13; count++)
   {
      printf("Month %d = $%.2f\n", count, expenses[count]);
   }
   return 0;
}
```

Output:
Enter expenses for month 1: **100**
Enter expenses for month 2: **200.12**
Enter expenses for month 3: **150.50**
Enter expenses for month 4: **300**
Enter expenses for month 5: **100.50**
Enter expenses for month 6: **34.25**
Enter expenses for month 7: **45.75**
Enter expenses for month 8: **195.00**
Enter expenses for month 9: **123.45**
Enter expenses for month 10: **111.11**
Enter expenses for month 11: **222.20**
Enter expenses for month 12: **120.00**
Month 1 = $100.00
Month 2 = $200.12
Month 3 = $150.50
Month 4 = $300.00
Month 5 = $100.50
Month 6 = $34.25

Month 7 = $45.75
Month 8 = $195.00
Month 9 = $123.45
Month 10 = $111.11
Month 11 = $222.20
Month 12 = $120.00

Exercise 2:
In this program you have to average 10 persons' individual score using a single array and a single for loop. Your output should be as follows:
Output:
Enter Person 1's score: **95**
Enter Person 2's score: **100**
Enter Person 3's score: **60**
Enter person 4's score: **100**
Enter Person 5's score: **25**
Enter Person 6's score: **0**
Enter Person 7's score: **85**
Enter Person 8's score: **85**
Enter Person 9's score: **95**
Enter Person 10's score: **85**
The average score is 73

Exercise 3:
This program declares and initialize an array **a,** in which each element has the same value as its subscript. i.e. a[i] = i. Add another loop that will multiply all the even numbers in the array by 3. Add a third loop to display all the values in the array on one line with a single space between each value.
Output:

```
0 1 6 3 12 5 18 7 24 9
```

Exercise 4:
Write a program that declares an array **A** and inputs **n** integer values in **A**. Then the contents of array **A** is copied to another array **B** in reversed order. Finally print the elements of array **B**.
Output:

```
Input the size of the array A: 8
Input the values of A: 10 20 30 5 22 11 44 78 55 66
Array B: 66 55 78 44 11 22 5 30 20 10
```

Exercise 5:
The *Fibonacci numbers* are numbers of an interesting sequence in which each number is equal to the sum of the previous two numbers. In other words,

$$F_i = F_{i-1} + F_{i-2}$$

where F_i refers to the ith Fibonacci number. By definition 1st Fibonacci number is 0 and 2nd Fibonacci number is 1.

$$F_1 = 0 \text{ and } F_2 = 1$$

Hence,

$$F_3 = F_2 + F_1 = 0 + 1 = 1$$
$$F_4 = F_3 + F_2 = 1 + 1 = 2$$

$F_5 = F_4 + F_3 = 1 + 2 = 3$ and so on.

Write a C program that will generate first n Fibonacci numbers. Do the following steps:

1. Declare an array of 100 integer values without initializing the elements.
2. Use two assignment statements to set the first two elements to be equal to 0 and 1 respectively.
3. Use a loop to set the values of the remaining elements, so that (apart from the first two elements) each element in the array is equal to the sum of the previous two elements in the array.

Output:

```
How many Fobonacci numbers you want to generate: 15
The series is:  0  1  1  2  3  5  8  13  21  34  55  89  144  233  377
```

Exercise 6:

Write a program that will find the smallest number from a list of unsorted numbers. Do the following steps:

1. Declare an array **A** of 50 integers
2. Input **n** number of integer values in A
3. Search for the smallest value among these n values in array A.
4. Display the smallest value

Output:

```
Input number of elements in the array: 11
Input values: 1959      278     25      36      568     5       98      45      87      96
The SMALLEST value is: 5
```

Exercise 7:

A is an m×n matrix and B is an n×p matrix too. Write a C program that multiplies these two matrices A and B to yield a new m×p matrix.

Output:

```
Input matrix A:
1       1       1
2       3       1
1       -1      -1
Input matrix B:
2       0       1
1       -1      0
3       1       -1
A X B:
6       0       0
10      -2      1
-2      0       2
```