

SHEET 7

- Write C++ program that reads an array of n integers in main. Then it calculates the binary representation of those integers in another array as follows:
 - a. Write function
 convertToBinary that
 takes the array of
 integers and returns
 another array of binary
 representations
 - b. Write function **Display** that displays both arrays.

<u>Hint</u>: to convert to binary, keep dividing the number by 2 (until it equals to 1) and store the remainder of each division as in this example: $(46)_{10}$ $(101110)_2$

```
46 0
23 1
11 1
5 1
2 0
1 1
```

```
Enter the number of elements

Enter 5 elements

2 6 8 12 3

The binary representation of each integer:

2 --> 10

6 --> 110

8 --> 1000

12 --> 1100

3 --> 11

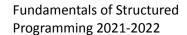
Press any key to continue . . .
```

- 2. Write C++ program that represents **tripDuration** as a *structure*. It consists of hours, minutes and seconds. The program should read n durations, then displays the total duration as follows:
 - a. Write function **Input** to read the durations

```
Enter the number of trips:

3
Enter 3 durations (Hours Minutes Seconds):
2 45 30
1 50 24
1 33 52
The total duration for 3trips is: (HH:MM:SS)
6 : 9 : 46
Press any key to continue . . .
```







- b. Write function calcTotal to calculate the total duration in (hours :minutes :seconds)
- c. Display result in main()
- 3. Write C++ program that accepts an array of n integers. Then, the program asks the user to type an integer value x and an index value i between 0 and n. Then value x should be inserted at index i in the array
 - a. Write function
 insertValue that inserts
 the value x at the place i
 in the array, by shifting
 each element right and
 dropping off the last
 element.
 - b. Input and output operations are carried out in main()

```
Enter the number of elements

Enter 5 elements

10 20 30 40 50

Enter the value to insert: 7

Enter index to place the value:2

10

20

7

30

40

Press any key to continue . . .
```



Faculty of Computer and Information Sciences Ain Shams University



Fundamentals of Structured Programming 2021-2022