



**Daffodil**  
*International*  
**University**

**Department of CIS**

**Subject: Structured Programming  
Fall 2020**

**Question**

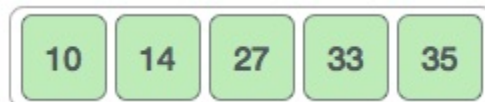
You are given a program where the user gives you **5 integer values in a random order**. However, you are expected to show them in **output in an ascending order** (smallest value first and then larger values following that)

For example, the user has given you these following values (that you store in an array)

–



The output is supposed to be –



This action is known as sorting. There are many different sorting algorithms that you might use to solve the given problems. However, **bubble sort** is the algorithm that you will be working with here.

**How Bubble Sorting works ->**

Bubble sort starts with very first two elements, comparing them to check which one is greater. In this case, value 33 is greater than 14, so it is already in sorted locations.



Next, we compare 33 with 27. We find that 27 is smaller than 33 and these two values must be swapped.



Next, we compare 33 and 35. We find that both are in already sorted positions.



Then we move to the next two values, 35 and 10. We know then that 10 is smaller 35. Hence, they are not sorted. We swap these values.

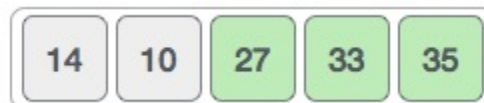


We find that we have reached the end of the array. After **one iteration**, the array should look like the picture above. For the next iterations, the final array should look like this -

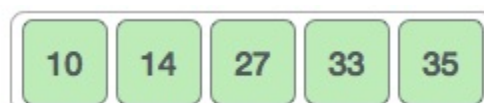
End of 2<sup>nd</sup> iteration:



End of 3<sup>rd</sup> iteration:



End of 4<sup>th</sup> iteration:



## Theory Part (35 Marks) –

### Task 1 (Design): 20 marks

Design a flowchart to find the solution to the problem above.

### Task 2 (Simulation): 10 marks

In this solution, you will have to implement a nested loop. How many times would the outer loop and inner loop run? Answer with the length of the given array being  $n$ .

### Task 3 (Critical Evaluation): 3 marks

Name three more sorting algorithms that can do this function as well.

## Lab Part (25 Marks) –

### Task 1 (15 Marks):

Write the C Code for the problem based on your algorithm which does not use any functions (only works in the main method).

### Task 2 (7 Marks):

Modify your solution in Task 1 and write a C Code that calls on a function **bubbleSort(array[], n)** where  $n$  refers to the length of the array and sorts the given array.

### Task 3 (3 Marks):

In your solution of Task 2, if you used temporary variables while swapping in the bubbleSort function, then replace that block of code with a **swap function** which would also do the same thing but in a different method.

## **Submission Guidelines:**

- Your submission should be in the form of a single word-processed document (**.doc** or **.docx**) that includes any necessary diagrams.
- Naming of the file as **example: 183-16-315.docx**
- **Marks will be deduced accordingly if any plagiarism of work is provided.**

**DEADLINE: 15<sup>th</sup> December, 2020**