

**Data Structures & Algorithms LAB**  
(BSCS-F18 Morning & Afternoon)  
**Lab # 10**

**Task # 1**

Implement a **recursive** C++ function which takes a character (**ch**) and a positive integer (**n**) and, prints a pattern on screen (see below). For example, it should display the following pattern when the arguments are '\*' and 4, respectively:

\*\*\*\*

\*\*\*

\*\*

\*

The prototype of your function should be:

**void printPattern1 (char ch, int n)**

**Hint:** Use the function **printChar** that you implemented in last lab.

**Task # 2**

Implement a **recursive** C++ function which takes an integer array (**arr**) and the starting (**start**) and ending (**end**) indices of that array, and prints all the values present in that array in a zig zag way i.e. initially the first and the last elements will be displayed, then the second and the second-last elements will be displayed, and so on. For example:

If the array contains {3,6,2,1,4} then the output should be 3 4 6 1 2

If the array contains {7,1,9,3,5,2} then the output should be 7 2 1 5 9 3

The prototype of your function must be:

**void printZigZag (int\* arr, int start, int end)**

**Task # 3**

Implement a **recursive** C++ function which takes an integer array (**arr**) and the starting (**start**) and ending (**end**) indices of that array, and returns the **sum of all elements** present in that array. The prototype of your function should be:

**int findSum (int\* arr, int start, int end)**

**Task # 4**

Implement a **recursive** C++ function which takes an array of integers (**arr**) and the starting (**start**) and ending (**end**) indices of a *portion* (part) of this array, and returns the **largest element** present in that portion of array *arr*. The prototype of your function should be:

**int findLargest (int\* arr, int start, int end)**

For example, the function call **findLargest(arr,3,6)** should determine and return the largest element present in the array **arr** between the indices **3** and **6** (both inclusive).

#### **Task # 5**

Implement a **recursive** C++ function which takes two integer arrays and their sizes as arguments, and determines whether these two arrays are equal or not. Note that two arrays are equal if they contain the same number of elements and the elements of both arrays occur in the same order. The prototype of your function should be:

**bool areArraysEqual (int\* a, int aSize, int\* b, int bSize)**

#### **Task # 6**

Implement a **recursive** C++ function which takes a c-string and its length as arguments and returns the number of vowels present in that c-string. The prototype of your function should be:

**int countVowels (char\* str, int length)**