

Data Structures & Algorithms LAB – Fall 2015
(BS-SE-F14 Morning & Afternoon)

Lab # 5

Instructions:

- **Solve the following tasks exactly in the given order.**
- Indent your code properly.
- Use meaningful variable and function names. Follow the naming conventions.
- Use meaningful prompt lines/labels for all input/output.

Note: All of the following tasks require you to write **recursive** functions. NO credit will be given for iterative implementation. Also write a main function to test the working of each function that you implement.

Task # 1

Implement a **recursive** C++ function which takes two integers **a** and **b** as arguments and returns their **product**. The prototype of your function should be:

int product (int a, int b)

Hint: Think about repeated addition.

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 # 3.1

Implement a **recursive** C++ function which takes a character (**ch**) and a positive integer (**n**) and prints the character **ch**, **n** times on the screen. The prototype of your function should be:

void printChar (char ch, int n)

For example, calling **printChar('Z',5)** should display **ZZZZZ** on screen.

Note: You are NOT allowed to use any loop in your function.

Task # 3.2

Implement a **recursive** C++ function which takes a character (**ch**) and a positive integer (**n**) and, 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 printPattern (char ch, int n)

Note: You are NOT allowed to use any loop in your function.

Hint: Use the function **printChar** that you implemented above in **Task # 3.1**.

Task # 4

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 # 5

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 # 6

Implement a **recursive** C++ function which takes two integers **num** and **den** as arguments and returns the **integer quotient** that will result when **num** is divided by **den**. The prototype of your function should be:

int quotient (int num, int den)

Hint: Think about repeated subtraction.

Task # 7

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 **index of the largest element** present in that portion of array *arr*. The prototype of your function should be:

int findLargestIndex (int* arr, int start, int end)

For example, the function call **findLargestIndex(arr,3,8)** should determine and return the **index** of the largest element present in the array **arr** between the indices **3** and **8** (both inclusive).

Task # 8.1

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 **reverse order**. The prototype of your function should be:

void printArrayReverse (int* arr, int start, int end)

Task # 8.2

Modify the **recursive** function that you implemented in **Task # 8.1** to take as its second argument the **size of the array (n)** instead of the starting and ending indices of the array. The prototype of this modified function should be:

void printArrayReverseNew (int* arr, int n)

Task # 9

Modify the **recursive** function that you implemented in **Task # 2** to take as arguments the **pointers** to the first and the last element of an integer array. The prototype of this modified function should be:

void printZigZagUpdated (int* pStart, int* pEnd)