BIL 142 LAB 5

Implement the recursive functions below.

- **int fibonacci(int number)**
  - ➔ The Fibonacci sequence fibonacci(1), fibonacci(2), fibonacci(3), ..., fibonacci(n), ... is defined by
    fibonacci(1) =1
    fibonacci(2) = 1
    fibonacci(n) = fibonacci(n-1) + fibonacci(n-2), for all n > 2
    This generates the sequence: 1, 1, 2, 3, 5, 8, 13, 21, ...
  - ➔ Write a function that computes the Fibonacci number of given integer input

- **int greatestCommonDivisor(int firstNumber, int secondNumber)**
  - ➔ This function takes two positive integer inputs and returns their greatest common divisor.
  - ➔ **Given Fact:** The greatest common divisor of two positive integers m and n for m<n is the same as the greatest common divisor of m and (n-m).

- **void checkPalindrome(char word[], int index, int length)**
  - ➔ A palindrome is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as **madam** or **racecar**.
  - ➔ The function takes a char array, a starting index and length of array as input.
  - ➔ The function must check whether input char array(string) is palindrome or not, then print result to the screen(i.e. "The entered word is a palindrome.")
  - ➔ Starting index will be **0**, i.e:
    char word[]="saippuakivikauppias"; //array length 19
    checkPalindrome(word, 0, 19);

- **void printSumTriangle(int A[] , int n)**
  - ➔ This function takes an int array and length of array as input.
  - ➔ Sum triangle is a triangle made by printing arrays which are made by sum of consecutive 2 elements of the previous array at each line in decreasing order of length of array.
  - ➔ This function must print a sum triangle such that the first level has all input array elements. From then, at each level the number of elements is one less than the previous level and

elements at the present level are the Sum of consecutive 2 elements in the previous level.

➔ **Input: A = {1, 2, 3, 4, 5}**

Output: [1, 2, 3, 4, 5]
[3, 5, 7, 9] -->(1+2=3, 2+3= 5, 3+4= 7, 4 + 5 = 9)
[8, 12, 16] -->(3 + 5 = 8, 5 + 7 = 12, 7 + 9 = 16)
[20, 28] -->(8 + 12 = 20, 12 + 16 = 28)
[48] -->(20 + 28 = 48)

Important: Add comments to your code and at the top of your code note which compiler you're using in a comment
Suggestion: Writing a main function all functions can be useful for you to check if the code is working(optional).

Upload your code file to 'uzak'.(deadline 21:10)