## Section 16

**PDS Lab** **Assignment - 6** **04.09.2018**

***Instructions:***

*Create a sub directory named as **Lab6**.*

*Give the name of the programs as <p>_1.c, <p>_2.c, .. etc. for the problem 1, 2….., respectively. Here <p> implies the part number. For example, Part-A*

*Store all the programs under this assignment in the directory **Lab6**.*
*Zip the entire directory **Lab6** and rename it as **<R>_Lab6_tar.gz**. where <R> denotes your Roll No.*
*You should upload your zipped file to the Moodle course web page latest by 11:55 hrs.*

## Part-A

1. The $n$-th ($n \geq 2$) Fibonacci number is defined as follow:

   $F_n = F_{n-1} + F_{n-2}$ with $F_1 = F_0 = 1$

   Read any integer number `x` from the keyboard and write a function `int checkFibonacci(int x)`, if it is an $n$-th Fibonacci number, and if so, then the value of $n$.

2. Binomial coefficient written as `C(n, r)` is the coefficient of the $x^r$ term in the polynomial expansion of $(1 + x)^n$. Binomial Coefficient can be recursively computed as
   `C(n, r) = C(n-1, r) + C(n-1, r-1)`, where $n \geq r \geq 0$. and `C(n, r)` = 1 if `r` = 0. Write a function `int ncr(int n, int r)`, which would calculate the value of `C(n, r)` recursively.
   [Hint: You should verify the calculation that $C(n,r) = \frac{n!}{r!(n-r)!}$ ]

3. The `n`-th Harmonic number is defined as $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$. Write a recursive function to calculate $H_n$, for any integer $n > 0$.
   [You should verify your calculation using the calculation of the sum of the series iteratively and writing another function for that.]

## Part-B

4. Two numbers are said to be co-prime, if the greatest common divisor of the numbers is one. For examples, 13 and 14 are co-prime but 14 and 21 are not. Write a C function `void CoPrime(int a, int b)` to test whether the pair of numbers a and b are co-prime. In the main program, read five numbers and use this function to test how many pair of them are co-prime.

[Hint: you should define `gcd(…)` to find the greatest common divisor of two numbers and `pair(…)` to find all the pairs from a given set of numbers.]

5. Read a set of *n* numbers and store them in an array. Write the following functions recursively.
   `int min(…)` : to return the smallest value in the array.
   `int max(…)` : to return the largest value in the array.
   `int sum(…)` : to return the sum of the values in the array.

## Part-C

6. The keypad of a phone is given below.


   Each numbered key (except 0 and 1) on the keypad has some characters corresponding to it. Typing a sequence of keys will enable us to compose a set of words. For an example, if we press keys 2-3-4 in this order, then the possible words that can be formed using this are

   *adg adh adi aeg aeh aei afg afh afi bdg bdh bdi beg*
   *beh bei bfg bfh bfi cdg cdh cdi ceg ceh cei cfg cfh cfi*

   How many different words can be formed when a user press a sequence of *n* ($n \geq 1$) keys? Print all the words on the screen corresponding to the sequence.

---*---

**Submission instruction**

Zip your ***Lab6*** directory. Upload your zip file into the Moodle server.