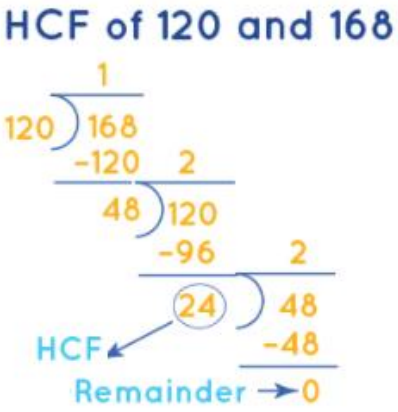
Data structures and Algorithms LAB – BSDSF21

(Morning and Afternoon)

Lab 03 – 24-01-2023

[OPTIONAL] the lats two questions are also mandatory, but are of 05 marks each, and the rest of the questions are of 18 marks each. So, in a sense is you can skip [OPTIONAL] questions, even then you can still get 90% percent marks.

1. Write a recursive function to compute the SUM of digits of a number **n** passed as it parameter. **HINT**: Use remainder and quotient of n divided by 10. Also, write a main function to test.
2. Write the recursive function ***printOctal(n)*** to print the octal number equivalent to its integer parameter. You have to write the main function which call the above-mentioned function in a loop to print first 50 decimal and corresponding octal numbers, one per line. **HINT**: Use remainder and quotient of n repeatedly divided by 8.
3. Function called ***sumover2*** that has one argument **n** which is an unsigned integer. The function returns a real type value which is described as, ***sumover2***(1) returns 1.0 and ***sumover2***(2) returns 0.5 as it is 1/2, ***sumover2***(3) returns 0.166667 as it is 1/2/3, and …..
4. Write the recursive function ***dec2oct(n)*** to return the octal number equivalent to its integer parameter. You have to write the main function which call the above-mentioned function in a loop to print first 50 binary numbers, one per line. **HINT**: Use remainder and quotient of n divided by 8, and later use to multiply the result by 10 on each recursive return. If n is ***438256*** the function will return an int data with value ***1527760***.
5. Function to return the ***greatest common divisor GCD*** (a.k.a. HCF) of its two parameters.



6. Implement and test the function to recursively compute e^x without using any extra computational function, but you may use/create power and factorial functions. [HINT: replace ∞ with a big number] [**OPTIONAL**]

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

7. Function to return the **n**th Fibonacci string. The Fibonacci strings are a series of recursively defined strings. F_0 is the string **a**, F_1 is the string **bc**, and F_{n+2} is the concatenation of F_n and F_{n+1}. For example, F_2 is **abc**, F_3 is **bcabc**, F_4 is **abcbcabc**, etc. [**OPTIONAL**]

********** The end **********