Introduction to Computer Programming and Data Structures
Assignment 02

Marks: 100
Bonus: 20 –for well indentation, variable name.

Submission Deadline: **2022-Aug-25**

Assignment # AP0201

- Write a program that computes and prints the nth prime number.

- Input: n

- Output: nth prime

- Hint: For $a, b \in \mathbb{N}$, if $a\%b == 0$, then a is divisible by b.

- Example: If input $n = 5$, the output will be 11 as the list of primes is [2, 3, 5,7, 11, ...] and 11 is the 5th prime.

[10]

Assignment # AP0202

- Write a program, given a positive integer n that computes SLP_n, the Sum of the Logarithms of all the Primes from 2 to some number n. Then it prints out the sum of the logs of the primes SLP_n, the number n, and the ratio of these two quantities, say $ratio_n$. Thus $ratio_n = SLP_n/n$, Test this for different values of n.

- Input: n

- Output: SLP_n n $ratio_n$ (separated by space)

- Use `math.h` library to compute logarithm. For compilation, use `-lm`. for example, `gcc -g -Wall -lm assignment_AP0202.c -o assignment_AP0204.out`. (`.out` for linux, `.exe` for windows)

[10]

Assignment # AP0203

- Write a program, print out the highest prime number HP_{uli} you can generate with considering only `unsigned long int` variable.

- Input: NA

- Output: HP_{uli}

[20]

Assignment # AP0204

- Handling input choices: Suppose you have the following functions,

  1. $area \leftarrow triangle\_area(a, b, c)$: It takes the length of three edges of a triangle and outputs the area of that.
  2. $length \leftarrow diag\_length(a, b, c)$: It takes edges of a rectangular cuboid, outputs the length of its diagonal.
  3. $ex\_val \leftarrow expo(x, y)$: Given two real numbers (float/double) x and y, it outputs $ex\_val = x^y$.

  Give the user four choices. Three choices to select the above function and one to exit. On user input, compute area/length/exponentiation and output the computed value. After each computation, the user will be given again four choices and continue until a choice for exit is chosen.

- Hint: To compute exponentiation, use `pow` function from `math.h` library

[20+10]

Assignment # AP0205

- Problem: Given a positive integer n and a real number (float) a, compute a^n without using `pow` function from `math.h` library.

- Input: $n\ a$

- Output: a^n

- Bonus: To compute a^n you may trivially require $n-1$ multiplication. How far can you reduce the number of multiplications?

- Hint: Consider binary representation of n i.e. $n = (b_1 b_2 \ldots b_k)_2$, for some k.

[10+20]