Lab session (16th Feb)

1. Easy

1.1. [Using `sprintf` function] Read integer `n` from `stdin`, and write the most significant 4 digits of `n` to a string using `sprintf`. Count the number of occurrences of each digit (0-9) in the string.
- Repeat till EOF and print final count

1.2. [2D-arrays and Matrix-matrix addition] Read 2 matrices from `stdin`. Each column is separated by ' ', and each row separated by '`\n`'. Print the sum of matrices.

2. Normal

2.1. [Generating random numbers and strings] The C stdlib supports different random number generators that can be used for various purposes.
- Read and understand the functionality of `rand()` and `srand()` from `stdlib.h`
- Generate a random number and a random string of length `n`, where `n` is an integer given as input to the program.

2.2. [Generating random entities] Create various entities using random numbers. Eg: IITH IDs, number plates of vehicles, Aadhaar numbers etc.
- Example IITH ID format: CCNNCCCCNNNNN, where C can be any char, N can be any digit from 0-9

2.3. [Generating random sentences] Print random sentences using words from /usr/share/dict/words. (For windows users, a dict file will be shared.)

2.4. [exp(x) using power series] Write a function to compute e^x by using the following formula of power series. Compare it with the value returned by `exp(x)` in `math.h`

$$\exp x := \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \cdots$$

3. Learn by Experiments

3.1. [Matrix routines] Write matrix-vector multiplication and matrix-matrix multiplication functions in C.

3.2. [atof] The `atof` function of stdlib converts strings to floats. Play with `atof` function to understand how it works.
- Also, give various varieties of floats as inputs and check the values it prints.
- For example, give very large and very small floats as inputs, and see its outputs. Check with the large limits given in `float.h`.

3.3. [Measuring time] Play with `time.h` created for measuring the time taken by sections of code. Create functions which iterate over 10^4, 10^5 and 10^6 values. Measure and report the time of these functions.

3.4. [Removing comments from C programs] Write a program to remove all C++ style comments from a C program. The output should be a file with no C++ style comments and the rest of the program be “intact”.

- Extend the above program to remove all C-style comments as well.
- The -E option of gcc (`gcc -E file.c`) does comment removal, macro substitution, inclusion of `#include` files and a lot more. You can compare the output of your program with the one generated by -E.