OBJECTIVES:

- To perform interactive input through keyboard and produce formatted output on screen
- To understand and perform string operations.

ASSIGNMENT PROBLEMS:

Task-1: Command interpreter: Write a C-program that reads a command in form of a string and produces the output according to the command. The program should incorporate the following three commands:

    a. `Sin x1 x2 Cos x1 x2 Tan x1 x2`: this command finds the Sin, Cos, and Tan of two numbers x1 and x2.

    b. `Log10 x1 x2 Exp X1 X2`: this command finds the Log10 and Exponential values of two numbers x1 and x2.

    c. `help` : this command prints available commands and their syntax (you can print the current list for the help command)

In case, the string entered by the user is not among the available commands, then the program should print the output of the `help` command.

Your program should generate a customized prompt to indicate that it is ready to take a command. For example, if your roll no is b10100, then your program should show the following prompt.

`b10100$`

An example command and its output is shown below:

`b10100$ sin 30 45 cos 30 45 tan 30 45`
`b10100$ 0.5 0.707 …..`
`b10100$`

Task2:
You are given with 5 protein sequences in the file "protein.faa". ".faa" is a standard format in which the protein sequences are stored for performing computations on them. Each line starts with a ">" symbol and reports the annotation of the protein, such as its function, cellular location, organism source, etc. The sequence starts from the next line after the annotation line.

Part 1: Read the sequences from the file one by one. Calculate the length of each protein sequence and store it in an array. Sort the array to determine the lengths of the smallest protein and the largest proteins. Report the following values on the STDOUT starting from smallest protein to the largest protein:

Annotation   length of protein

You can use the following code to read from a file:

```
#include <stdio.h>
#define MAX_STRING_SIZE 1000
int main()
```

```
{
    char buffer[MAX_STRING_SIZE];
    FILE *f = fopen("textFile.txt", "r");
    fgets(buffer, MAX_STRING_SIZE, f); /* reads first line, and changes file pointer so that it points to next
line */
    printf("Line read: %s\n", buffer);
    fclose(f);
    return 0;
}
```

Optionally you may input the sequences from the STDIO. Input the annotation line and then the entire sequence in one line.

Part 2: Use the following simple pattern to search in each of the five sequences:

VTEVGIFTPKAVGR

Print on STDOUT the annotation of each sequence and the start and end location of the pattern where it occurs in this sequence. Repeat this for all the five sequences.

Sample Faa file:
>CCO74927.1 GTP binding proteins [Streptococcus agalactiae] **(This is the annotation line of 1ˢᵗ protein sequence)**

MTFKSGFVAILGRPNVGKSTFLNHVMGQKIAIMSDKAQTTRNKIMGIYTTETEQIVFVT **(This is the sequence line 1 for first protein sequence)**

EVGIFTPKAVGRDFMVESAYSTLREVETVLFMVPADEKRGKGDDMIIERLKAAKIPVIL **(This is the sequence line 2 for first protein sequence)**

>>CCO74532.1 GTP binding proteins [Streptococcus agalactiae] **(This is the annotation line of second protein sequence)**

MNIEDLKKRQEKIRNFSIIAHIDHGKSTLADRILEKTETVSSREMQAQLLDSMDLERER **(This is the sequence line 1 for second protein sequence)**

GITIKLNAIELNYTAKDGETYIFHLIDTPGHVDFTYEVSRSLAACEGAILVVDAAQGIE **(This is the sequence line 2 for second protein sequence)**

Important note: You must concatenate all the sequence lines for each protein sequence into one string and then perform the asked operations.

NOTE:
- Each task carries 2 marks
- You are required to bring pseudo codes (and not full programs) for each of the tasks written in a notebook to the lab session and present them to the Instructors/TAs for evaluation
- The codes need to be created from scratch while you are in lab

REMEMBER:
- To use spaces and indentation to improve the readability of your code
- To add a comment block at the top of your code file stating your name, roll no., assignment and task no.
- To provide comments at appropriate places to aid lucid comprehension of modules