Term Project

General Rules for the Project

Teams:
You are to work in teams of two students. Each team will work on the project, do the design, implementation, testing, and prepare a final report. Both members of the team are expected to participate equally in all aspects of the project. Both members of the team will receive the same grade on the project.

Project Requirements:
The team will choose ONE project among the suggested projects in this document. Alternatively, the team can propose their own project based on their own research interests, with the approval from the instructor. In this case, a detailed description of the programming project will have to be submitted by the team to the instructor by October 23, 2019, for approval.

Your final project grade will be computed in the following way:

b) Project report: 80% c) Presentation: 20%

Important Dates:

a) Project Proposal
October 23, 2019: Each team submits in Canvas a one-page description of the project. If the topic is not one of the suggested projects described in this document, then the team includes a detailed description of their proposed project.

b) Project Report
December 2, 2019: Printed copies of the final project are due at the beginning of the lecture. Do not forget to include all pertinent documents. Submit all files described under "Submission Requirements" below in Canvas.

c) In-Class Presentation
December 2 or 4, 2019: Each group gives a 10-minute, in-class presentation (5 minutes per student).

Submission Requirements:

A) Submit a hard copy of the project containing all of the following:

1) Title page.
2) Table of contents (with page numbers).
3) An essay of not more than 6, and not less than 4, 1 and 1/2 spaced pages (font size: 11 or 12 points) describing the problem, the overall organization, design of your program, and

results of your project. The essay should give the user an overall roadmap of your code. Include a flowchart or an UML diagram to show the design of your program. Do not forget to number pages!

4) Include one sample output of your program. Make sure that your output is readable and well formatted.
5) Instructions for running your program.
6) Optional: A one-page description of the choice of the test data (in addition to the ones specified in the project) and the testing strategy you used.

B) Submit files to Canvas (one submission per group):

a) The source code (fully documented)
b) The input files (if any)
c) The document specified in part A

PROJECT CHOICES

A) DNA ANALYSIS SYSTEM:

Write an object-oriented program for analyzing DNA sequence, including CG content, k-mer counting, complementary sequence, and translation of all six-reading frames into protein sequences. For translation example, check out the ExPASy Translation tool: https://web.expasy.org/translate/. Design this system using UML first before implementing it in Python.

B) CRISPR GENE EDITING SYSTEM:

Design CRISPR gRNAs to suppress antimicrobial resistance in MSRA by disrupting the penicillin-binding protein (mecA) gene.
Background: http://clsjournal.ascls.org/content/30/4/207
S. aureus mecA gene sequence: https://www.ncbi.nlm.nih.gov/nuccore/KF058908.1
CRISPR gRNA design considerations:
https://blog.addgene.org/how-to-design-your-grna-for-crispr-genome-editing

C) PAIRWISE SEQUENCE ALIGNMENT:

Implement the dynamic programming algorithm for generating an optimal global pairwise alignment of any two given DNA sequences. Here is a tutorial on how it works:
http://avatar.se/lectures/molbioinfo2001/dynprog/dynamic.html

D) STUDENT MANAGEMENT SYSTEM:

Write an object-oriented program for managing students in CS22B class. This system helps keep track of project team info, assignments, team project report, and exams. Using this system, instructor can easily update or query information for any student. Design this system using UML first before implementing it in Python.

E) THE GAME OF NIM:

Develop this game using an object-oriented paradigm. In this game, two players alternately take marbles from a pile. In each move, a player chooses how many marbles to take. The player must take at least one but at most half of the marbles. Then, the other player takes a turn. The player who take the last marble loses. Write an object-oriented program in which the computer plays against a human opponent. Generate a random integer between 10 and 100 to denote the initial size of the pile. Generate a random integer between 0 and 1 to decide which player takes the first turn. Generate a random integer 0 and 1 to decide whether the computer plays *smart* or *dumb*. In *dumb* mode, the computer simply takes a random legal value (between 1 and n/2) from the pile whenever it has a turn. In *smart* mode, the computer takes off enough marbles to make the size of the pile a power of 2 minus 1 – that is, 3, 7, 15, 31, or 63. That is always a legal move, except if the size the pile is currently one less than a power of 2. In that case, the computer makes a random legal move. When implementing this program, be sure to have classes Pile, Player, and Game. Design this system using UML first before implementing it in Python.

F) MUSICAL PROTEINS PROJECT:

This is a challenging problem that will require you to do some very interesting reading and understanding, especially if you are new to music. Over the years, researchers have acknowledged the fact that there are significant similarities between the structures of proteins and music. Both structures are composed of phrases organized into themes [1]. Write a program that converts amino acid sequences to music. By mapping amino acids to musical notes, one can play and listen to proteins. A quick search on the internet will yield a few mappings that have been introduced in the past. One challenge discovered by researchers is generating pleasant music while preserving relationships between amino acids properties. Test your program on short protein sequences, such as the beta globin protein:

```
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAV
MGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNV
LVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH
```

1] Dunn J. Clark M.A. (1999) Life Music: Sonification of Proteins. Leonardo V.32: 25-32.