**Take Home Mid-Term Spring 2024**

Due on 11:59pm Feb 25, 2024, by e-mail to florin@cmu.edu

Your Name: ……………………………………………

Coding Team colleagues (team of up to 4 members sharing work on the same code) in the order of your collaboration with them (the most collaboration on top):

…………………………..

…………………………..

…………………………..

**Rules**:

* submit one document per student
* submit one code archive per team
* use any help available, including me, people in the team, other people you know, ChatGPT-like interfaces, online documentation, etc.

**Problem to solve** (ambiguous, management style formulation):

Find groups of 2 or more consecutive characters (as many as possible, including spaces or symbols) which can be found in at least 60% of the entries in a column of a csv file.

**Deliverables**:

1. This document filled individually by every student; some of the answers may be shared.

2. Archive of the code; each module in a separate file.

**Suggested structure of the solution (modules):**

Modify if your structure is different.

User: yes, it is useful to consider the user as a module for documentation purposes

* Input: command line options, e.g. data file name, percentage of string matches (default 60%), minimum number of characters in the match (default 2)
* Output: presentation and meaning of the results

Kernel:

* Input: extract information from the command line options
* detect data file format
* call the other modules as needed
* Output: format and display results

Data extraction:

* Input: read the data file
* Output: return the columns as separate entities

Find matching substrings for the entries in a column:

* Input: receive a column
* iterate the search process until no more matches are found
* Output: return the results

**Assumptions:**

State unambiguously any assumptions you made to solve the problem (ask me for clarifications if the original formulation doesn’t say much to you).

Motivate your assumptions if needed.

**Code documentation**:

1. Draw the diagram of the modules. List APIs describing precisely the communication channels (input and output) between modules.

2. Which modules should be optimized to increase the sequential execution speed? Present your thoughts about this subject, no implementation needed.

3. Briefly describe the algorithm for finding matches.

If a data column has N lines, how do you expect the execution time to depend of N?

4. Copy&paste the result of your code using the sample CSV file provided with the exam.

5. Instructions (assume command line): Steps to follow by an user to go from your code archive to the listed results (copy&paste from the terminal).

**Hypothetical parallel implementation:** (no need to actually produce a parallel implementation)

Comments about which part of the code is suitable for parallel execution and why.

What type of partitioning did you consider to use?

**Credits**: (not considered for grading)

List of people (even outside your team) and ChatGPT-like interfaces who helped you with your work, ordered by how helpful their contribution was (first one was the most helpful).