*Data Structures Course Project*

*Data Science and AI Department*

*Deanship of Engineering and Intelligent Systems*

*UCAS*

# Project: The Text Processor

**Objective:**

This project aims to explore the application of various data structures to solve text processing problems. You will implement a program that performs a set of operations on a given input text, utilizing arrays, dictionaries, sets, stacks, queues, and optionally, linked lists.

**Requirements:**

1. **Input:**
   - The program should begin by prompting the user to enter a string of text.
2. **Character Analysis:**
   - **Store the input string in an array of characters.**
   - Determine and display the frequency of each character in the input string using a dictionary.
3. **Word Analysis:**
   - Split the input string into individual words.
   - Store the unique words in a set.
   - Determine and display the frequency of each word in the input string using a dictionary.
4. **Stack Operations:**
   - Implement a stack data structure.
   - Push each character of the input string onto the stack.
   - Pop characters from the stack and print them to create a reversed version of the input string.
5. **Queue Operations:**
   - Implement a queue data structure.
   - Enqueue each word from the input string into the queue.
   - Dequeue words from the queue and print them to create a "first-in, first-out" ordering of the words.
6. **Linked List (Optional):**
   - Create a simple linked list to store the words in the order they appear in the input string.
   - Implement basic linked list operations (insertion, traversal).

**Note:** Feel free to use other and more advanced data structures from what we learned like Trees, or other data structure that convenience and provides efficiency.

**Output:**

- The program should display the following:
    - Character frequencies.
    - Word frequencies.
    - Reversed string (using the stack).
    - "First-in, First-out" ordering of words (using the queue).
    - (Optional) Linked list traversal (displaying words in the order they were stored).

**Deliverables:**

- Python code implementing the described functionalities.
- A brief report documenting your **design choices** (*optional*), **implementation details**, and any **challenges encountered**.

**Evaluation Criteria:**

- Correctness of the implemented functionalities.
- Efficiency and clarity of the code.
- Proper use of data structures.
- Code readability and documentation.
- Overall project quality and presentation.

**Bonus Points:**

- Implement additional text processing features (e.g., word count, search/replace).
- Allow users to choose which data structures to use for specific tasks.

**Note:** This project provides a framework. Feel free to explore and add your own creative touches to enhance the project and deepen your understanding of data structures.