PRACTICAL EXAM – CSD203 – Spring 2024

Duration: 60 minutes

Requirements:

- Students are allowed to use all learning materials such as textbooks, slides, and lab exercises.
- Students are NOT allowed to use the Internet during the exam.
- Students are allowed to use IDEs such as PyCharm, Visual Studio Code, etc.
  However, AI-assisted extensions such as TabNine or Copilot are prohibited.
- Students are NOT allowed to submit any personal information in the exam file such as name, student number, etc.
- Students must compress all of their work into a file named CSD203_PE.zip.

The exam files include this document, 2 python files named linkedlist.py and graph.py.

Question 1:

Use the **linkedlist.py** file. Complete the following requirements:

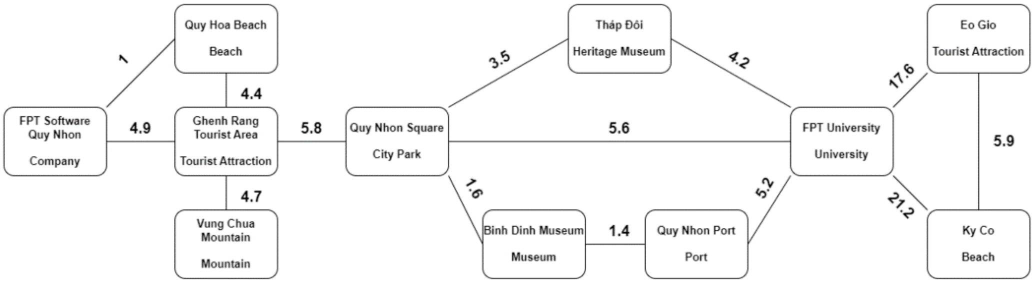
- Create a ***Location*** class that includes the following properties: name, category, address, and description. The category, address, and description are default to None. (1.0 pt.)
- Complete the following methods of the ***LocationLinkedList*** class:

No.	Methods	Descriptions	Marks
1	add_location	Add a location to the beginning of the linked list.	1.0 pt.
2	remove_location	Remove the first location in the linked list have this name, return the location. Otherwise, return None.	1.0 pt.
3	search_location	Finds all locations in the linked list whose names contains the keyword.	1.0 pt.
4	display	Show all the locations by their names in the linked list.	1.0 pt.

Question 2:

Use the **graph.py** file. Complete the following requirements:

- Implement the ***insert_location, insert_path*** methods and create a graph as the image below. (2.0 pt.)



(Nodes contain names and categories)

- Complete the ***dijkstra*** method using the instructions included in the file. (1.0 pt.)

- Implement ***shortest_path*** method to find the shortest path between two points in the graph. The output should be a linked list of locations on the shortest path between those two points, or None if two locations are not connected. (0.5 pts.)
- Complete the ***prim_jarnik*** algorithm to find the minimum spanning tree in a graph starting from one point, using the instructions included in the file. (1.0 pt.)
- Implement the ***mst_path*** algorithm to find the path between two points in the graph using the MST. The output should be a linked list of locations on the MST path between those two points, or None if two locations are not connected. (0.5 pts.)

END