Recall the data structure that you designed last week for storing grades of lab groups. In this example class, we are about to develop a grading system consisting of the following functionalities:

* Input individual grades, e.g., the grade of student no. 1 in FE1 is 90.
* Query the grade of a student in a lab group
* List all the grades in a lab group
* Get the highest grade in a lab group

**Discussion 1**

Develop a Python function inputRecord(dataBase, group, id, score) for TAs to enter one record, where dataBase is the database implemented by your data structure, group is a string representing a group name, id is a student’s id number (positive integers ranging from 1 to 40), and score is the grade of the student.

**Discussion 2**

Develop a Python function query(dataBase, group, id) for TAs to get the score of a student in a lab group, where dataBase is the database implemented by your data structure, group is a string representing a group name, and id is the student’s id. This function should return the score.

**Discussion 3**

Develop a Python function listGrades(dataBase, group) for TAs to get all the student grades in a group, where dataBase is the database implemented by your data structure and group is a string representing a group name. This function should return a list of grades in the group.

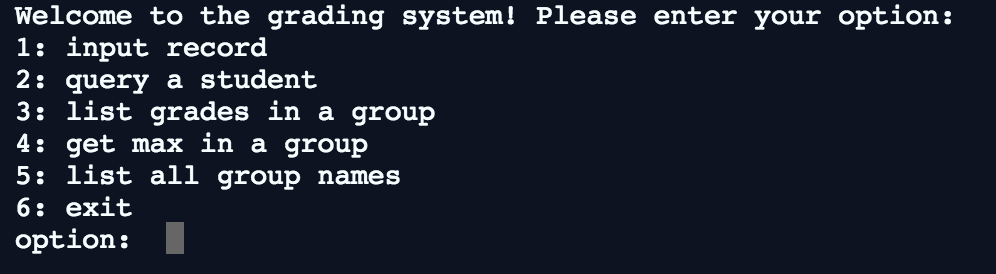
**Discussion 4**

Develop a Python function maxGrade(dataBase, group) for TAs to get the highest grade in a group, where dataBase is the database implemented by your data structure and group is a string representing a group name. This function should return the highest grade.

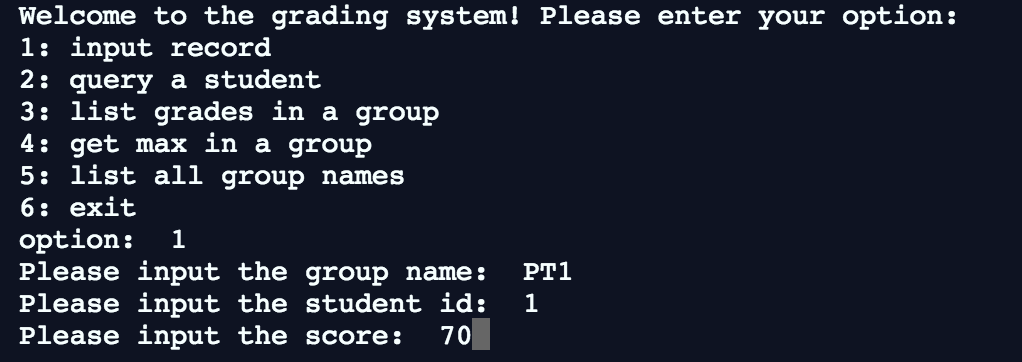
**Discussion 5 (Optional)**

Develop the main program to show the options to users, get the option, and invoke the corresponding functions developed.

For example, show the available options and get the option from users:



For example, invoke the corresponding function:



Guide for Instructor

**Discussion 1**

* The purpose of this question is to let the students practice defining functions in Python. Users have to provide at least three arguments: the group name, the student id, and his/her grade.
* The implementation should correspond to the data structure that they designed in the previous example class.
* Example code is provided for reference.

**Discussion 2**

* This function is for querying the grade of a student in a group. Users have to provide at least two arguments: the group name and the student id.
* Notice that the implementation may vary due to different data structures that they designed in the previous example class.
* Users may have invalid arguments. Tell the students to focus on valid cases first. Do the improvements as homework.
* Example code is provided for reference.

**Discussion 3**

* This function is for listing all the grades in a group. Users have to provide at least one argument: the group name. You can encourage students to use list comprehension to implement this function.
* Notice that the implementation may vary due to different data structures that they designed in the previous example class. Again, users may have invalid arguments. Tell the students to focus on valid cases first. Do the improvements as homework.
* Example code is provided for reference.

**Discussion 4**

* This function is for getting the highest grade in a group. Users have to provide at least one argument: the group name. Students can reuse their code in the previous example class. They just to wrap their code in a function.
* Example code, 1A2B.py, is provided for reference.
* Example code is provided for reference.

**Discussion 5 (Optional)**

* This question is to let the students test their functions.
* Example code is provided for reference.