Week 7 Exercises

Exercise 7.1: Student Record Management

Instruction: Name your main class StudentRecordManager.

Write a program that allows the user to manage student records. The program will have the following features:

1. Add a new student
2. Modify student's information
3. Delete a student
4. Display the list of students
5. Exit

Add a new student: Add a new student to the system. Since student ID is unique, your program has to check for duplicated ID before adding the new student to the system.

Update student's information: Given a student ID, this option allows the user to update student's first name, last name and score.

Delete a student: Given a student ID, mark that student record as deleted (update student's ID to 9999999).

Display the list of students: Show all student records. Note that deleted records will not be shown here.

Exit: Terminate the program.

Exercise 7.2: N-Grams

Instruction: Name your class NGramExtractor.

In the computational linguistics field, an n-gram is a contiguous sequence of n items from a given text or speech. Items could be syllables, letters or words depending on the application. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram"; size 3 is a "trigram"

In this exercise, you will implement a method that extracts n-grams (in lowercase) from the given text where one unit is a word. Use the following method signature:

```
String[] extractNGrams(String text, int n)
```

For instance, if the text is "To be or not to be",

- `extractNGrams(text, 1)` will return `["to", "be", "or", "not", "to", "be"]`.
- `extractNGrams(text, 2)` will return `["to be", "be or", "or not", "not to", "to be"]`.
- `extractNGrams(text, 3)` will return `["to be or", "be or not", "or not to", "not to be"]`.

Hint: You need to tokenize the text first.
Reference: The definition of n-gram is from wikipedia.