**Week 6 Extra In-Class Exercises (More on Lists)**

**Q1: Books [ \*\* ]**

(Download Week 6 Extra In-Class Exercises Starting Code from eLearn.)

**Part (a)**

You are given a list of tuples, where each tuple represents a copy of a book in a library’s collection. Specifically, each tuple consists of four elements:

(1) A string representing the title of the book.

(2) A string indicating the edition of the book. E.g., the third edition of a book would have this string set to “Ed-3”.

(3) A string indicating whether this book is a hardcover book or a paperback book, i.e., the value of this string is either “hardcover” or “paperback”.

(4) An integer indication the number of copies of the book with the specified title, edition and type (hardcover vs. paperback).

Note that the library has multiple books having the same title.

Write a function called get\_unique\_titles() that takes in such a list of tuples as its parameter. The function returns a new list of strings, which are the unique book titles from the original list.

For example, get\_unique\_titles([("Intro to Programming", "Ed-2", "paperback", 2), ("Intro to Python", "Ed-1", "paperback", 5), ("Intro to Programming", "Ed-3", "hardcover", 4)]) should return the list ["Intro to Programming", "Intro to Python"].

**Part (b)**

Write a function called get\_titles\_and\_counts(). The function takes in the same kind of list of books as described above. The function should return a list of tuples, where each tuple contains two elements: (1) A book title, and (2) the number of copies of that book.

For example, get\_titles\_and\_counts([("Intro to Programming", "Ed-2", "paperback", 2), ("Intro to Python", "Ed-1", "paperback", 5), ("Intro to Programming", "Ed-3", "hardcover", 4), ("Intro to Python", "Ed-3", "hardcover", 3)]) should return the list [("Intro to Programming", 6), ("Intro to Python", 8)].

We get this result because there are six copies of “Intro to Programming”:

* Two in a 2nd edition in paperback and
* Four in a 3rd edition in hardcover

and there are eight copies of “Intro to Python”:

* Five in a 1st edition in paperback and
* Three in a 3rd edition in hardcover.

**Q2: Retrieve Numbers [ \*\*\* ]**

(Download numbers\_test.py from eLearn)

Inside **numbers\_test.py**, define a function called retrieve\_numbers(). The function takes in a string that contains some numbers as its parameter. The function **returns** the sequence of these numbers separated by spaces as a string.

For example,

* retrieve\_numbers("12abc600$##0900AB 100X")

returns the string "12 600 0900 100"

* retrieve\_numbers("34.5689abc980")

returns the string "34 5689 980"

* retrieve\_numbers("xyz")

returns the string ""

* retrieve\_numbers("abc25xyz")

returns the string "25"

**Q3: Common Modules [ \*\*\* ]**

(Download common\_modules.py and common\_modules\_test.py from eLearn.)

Define a function called check\_common\_modules() inside the file common\_modules.py. The function takes in the following parameter:

* a list of tuples called student\_list: This list contains a sequence of students. Each student is represented as a tuple containing a name and a list of strings, representing the modules this student is taking this term.

E.g., ('Mike', ['IS110', 'IS111', 'STAT101', 'GMGT003']) represents a student called Mike who’s taking four modules this term: IS110, IS111, STAT101 and GMGT003.

The function returns True if *every* student in the list shares a common module with *either* the previous student in the list or the *next* student in the list; Otherwise, the function returns False.

Note:

* For the first student in the list, he/she has to share a common module with the second student. For the last student in the list, he/she has to share a common module with the last but one student.
* If the list contains only one student or is empty, the function returns False.

For example,

* if student\_list is  
   [('Zeus',['IS01', 'IS02', 'CS03', 'STAT01']),  
   ('Apollo', ['IS02', 'CS04', 'ECON02']),  
   ('Athena', ['CS10', 'CT07', 'STAT10']),  
   ('Ares', ['CS11', 'CT03', 'STAT10'])]  
  then check\_common\_modules(student\_list) returns True. (Zeus and Appollo share a common module 'IS02', Athena and Ares share a common module 'STAT10'.)
* If student\_list is  
   [('Zeus',['IS01', 'IS02', 'CS03', 'STAT01']),  
   ('Apollo', ['IS02', 'CS04', 'ECON02']),  
   ('Athena', ['CS10', 'CT07', 'STAT10']),  
   ('Hades', ['ECON02', 'IS01', 'STAT01']),  
   ('Hera'), ['STAT01', 'ECON02', 'IS02']]  
  then check\_common\_modules(student\_list) returns False. (Athena does not share any common module with either Apollo or Hades.)

Use the provided **common\_modules\_test.py** to test your code. DO NOT modify **common\_modules\_test.py**.