Quiz #3

Instructions

Customers Who Bought This Item Also Bought

The goal of this assignment

- build a very simple recommendation system
- use PySpark to solve the problem
- use **./bin/spark-submit** (which is used in production environments)

Problem description: What to do

The goal is to build a very simple recommendation engine based on the concept of Amazon's Customers Who Bought This Item Also Bought (CWBTIAB). This concept is explained very well in this article: **https://measuringu.com/affinity-analysis/**

Input File and format:

The input to your PySpark program will a text file. Your program has be a general program
to accept any input as a text file. There can be duplicate records, which must be removed by your PySpark program.

Format of each record:

<user><:><book>

Input Example:

```
$ cat purchases.txt
u1:book1
u1:book2
u1:book2
u1:book7
u2:book3
u2:book2
u2:book2
u2:book7
...
```

Actual input

The actual input will be provided later. When submitting your solution, you must use the actual input (and not the sample input). Sample input is provided for your debugging and testing purposes.

Expected output:

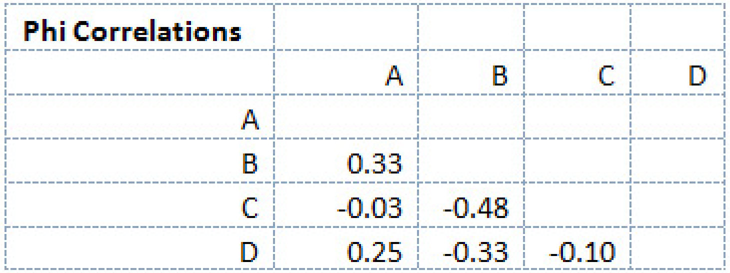
The goal is that when a user is looking at book X, we should give a recommendation like:

Customers Who Bought This Item X Also Bought: X1, X2

At most 2 items will be recommended.

Therefore, your output has 2 parts:

Part-1: a sparse similarity matrix (more than half will be empty). See the following image: we have 4 books and similarities have been calculated for every pair of books. Note that similarity(A, B) is the same as similarity(B, A). For efficiency, you have to calculate one of them. Also, you do not need to calculate similarity(X, X) for any X.



Phi Correlations				
	A	B	C	D
A				
B	0.33			
C	-0.03	-0.48		
D	0.25	-0.33	-0.10	

Part-2: generate a recommendation for all books (top-2, in descending order)

```
A: B, D
B: A, D
C: A, D
D: A, C
```

Therefore, if a customer is looking for A, then we will recommend B and D.

Documentation of your program:

- all steps must be explained
- a high-school student should understand what is happening in your program
- you must use RDDs (no DataFrames)
- Minimize the Verbosity of Spark (see: https://stackoverflow.com/questions/28189408/how-to-reduce-the-verbosity-of-sparks-runtime-output)