```
Program 2 pgm2cop3530dsA  - cop3530 Data Structures and Algorithms

Professor: Michael Robinson
e-mail   : mrobi002@cs.fiu.edu
Web Page : www.cs.fiu.edu/~mrobi002/teaching

- Program must be named: yourLastNameFirstLetterOfYourFirstNamepgm2.java
  If your name is George Washington the program should be named:
                         WashingtonGpgm2.java

- Turn in the signed source code on paper, and email me the source code.

- Make sure the program is properly documented and aligned uniformally, looking professionally,
  I will take points off if it is not.

- Include the following header in every program:

/*********************************************************************
Author     : Your Name
Course     : COP 3530 Date, Time and place of class
Professor  : Michael Robinson
Program #  : Program Purpose/Description
             {A brief description of the program }
Due Date   : MM/DD/YYYY

Certification:
I hereby certify that this work is my own and none of it is the work of any other person.
 ..........{ your signature }..........

*********************************************************************/


      Purpose of this program: Implement chapter 3 && 4 ideas


***************************************************************************
*  Make sure to use the implementations shown in the Data Structures Book  *
*  DO NOT use the java build commands, except for the Queues and Stacks    *
***************************************************************************


Chapter 3

1 - Worth 1 point
Implementation of LinkedList
  - Create a double linked LinkList
  - Add "I", "did not", "Like", "Programming"
  - Print the linklist
  - Remove "did not"
  - Between nodes containing "I" and "Like"
    Insert "love"
  - Print the linklist


2 - Worth 1 point
The Stack ADT
  - Create a Stack
  - Add "Joe", "is", "taking", "java", "programming"
  - Print the top value in the stack
  - Print "taking", "java", "programming" from the stack


3 - Worth 1 point
The Queue ADT
  - Create a Queue
  - Add "Joe", "is", "taking", "java", "programming"
  - Print the top value in the Queue
  - Print "taking", "java", "programming" from the Queue
```

Chapter 3 Trees
4 - Worth 1 point
Binary Search Trees
- create a binary search tree with the following data
  abc, 1, two, _james, 78, 34, -98
- findMin
- findMax
- insert Camilo
- remove two

5 - Worth 2 point
AVL Trees
- Create an AVL tree
- Insert 1, 2, and 3 and balance it as in page 127 graph "after" below fig 4.33
- Print tree
- Insert 4
- Print tree
- Insert 5
- Balance tree
- Print tree

6 - Worth 2 point
Implementation of Maps
- Using a map data structure, and the following data

sandals	2.50
T-shirt	3.75
hat	5.50
sun-cream	9.75

add the previous data into a map
them access all data from the map,
print the data in the map, one line per record,
add the amounts of products and print the total
calculate 7% tax of the purchase
and print the total amount to be paid