WIA1002 DATA STRUCTURE
WEEK 7 LAB

Do not use any existing Collection classes available from the Java API Library.

1. Create a generic MyQueue class that has the following methods.

Method Name	Specification
Constructor	Default constructor
isEmpty	Return whether or not the queue is empty
peek	Return the value of the first element in the queue
enqueue	Add an element to the bottom of the queue
dequeue	Remove element from the top of the queue
display	Display all elements in the queue

2. Using queue structure, extract the needed elements in a String input and store them. Let "D" be depositing and "W" be withdrawing. Assuming that the balance amount is 1000. Display the String in the following format.
   Use this as the input: D 400 | W 300 | W 700 | D 450 | W 120

```
Enter transactions : D 400 | W 300 | W 700 | D 450 | W 120
D 400 --> W 300 --> W 700 --> D 450 --> W 120  --> Initial Balance: 500
BUILD SUCCESSFUL (total time: 7 seconds)
```

3. Given a text file named lab7q3.txt, read it and store its contents in a few queues. After that, display the elements according to the order.

```
Product Code in Queue : P03 --> P02 --> P04 --> P09 -->
Product : P03
happy --> worried --> sad -->
Product : P02
Supra --> Miata --> Ftype -->
Product : P04
water --> earth --> fire -->
Product : P09
watermelon --> apple -->
BUILD SUCCESSFUL (total time: 1 second)
```

4. Simulate a simple card game with two players. There are four colors in total, with precedence of Red > Green > Blue > Yellow. There are 10 cards for each color with precedence of 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3 > 2 > 1. Randomly assign 5 cards for each player and compare their precedence, the bigger card scores one mark for the player. At last, compare the total scores of two players and announce the winner.

```
Player 1 Card
Seven Yellow --> Eight Yellow --> Three Green --> Eight Yellow --> Eight Red -->
Player 2 Card
Two Red --> Two Green --> Six Yellow --> Three Yellow --> Eight Green -->
Player 1 Score: 3
Player 2 Score: 2
Player 1 wins!
BUILD SUCCESSFUL (total time: 13 seconds)
```

5. An investor is trying to calculate the profit gained. The investor always sells the stock that is being held for the longest. For example,
   Day 1: Bought 100 shares at $20
   Day 2: Bought 20 shares at $24
   Day 3: Bought 200 shares at $36
   Day 4: Sold 150 shares at $30
   Total Profit: $940

   The formula for the above calculation is (100 * (30-20)) + (20 * (30-24)) + (30 * (30-36)) = 940. Use MyQueue to implement this process.

6. Steve is trying to implement an AI system that can identify the type of vehicle and arrange them according to their priority to pass through a passage that merge from a few roads. Construct a vehicle class that defines their type and priority precedence. Modify the enqueue or dequeue method so that they are ordered according to their priorities.

```
Car             1  (Priority=1)
Truck           2  (Priority=2)
Motorcycle      3  (Priority=0)
Motorcycle      4  (Priority=0)
Truck           5  (Priority=2)
Car             6  (Priority=1)
Truck           7  (Priority=2)
Truck           8  (Priority=2)
Motorcycle      9  (Priority=0)
Car             10 (Priority=1)

Truck           2  (Priority=2)
Truck           5  (Priority=2)
Truck           7  (Priority=2)
Truck           8  (Priority=2)
Car             1  (Priority=1)
Car             6  (Priority=1)
Car             10 (Priority=1)
Motorcycle      3  (Priority=0)
Motorcycle      4  (Priority=0)
Motorcycle      9  (Priority=0)
BUILD SUCCESSFUL (total time: 1 second)
```