**Problem 1.** [5pts]

Transform the following expressions into postfix notation:

- \((A + B) \times (C + D)\)
- \((A \times (B + (C / D))\)
- \((A + (B - C)) + D \times (E - G)\)
- \(((A + B) \times (C - D)) / (E - G) \times F\)
- \(((A \times B) + C) / (A - C \times D)\)

Submit your answer in a plain text file called Prob1.txt.

**Problem 2.** [25pts]

Implement a generic class `ArrayQueue` that implements the Queue Abstract Data Type using arrays. It should allocate an array of some reasonable initial size and allocate a new array of double size anytime the old array overflows. You do **not** have to shrink the array when the queue gets very small. The class should be generic. As you know, Java does not allow allocation of generic arrays. To allocate an array of generic type \(T\), you have to allocate an array of type `Object` and then case it to `\(T[]\)`, as follows:

```java
T[] array = (T[]) new Object[10];
```

Ignore the warning that will be generated by the compiler. Create a simple driver class that tests the implementation of the Queue by using `enqueue` and `dequeue` operations and printing the output of the queue (the item that you get back from the `dequeue` operation). Submit all your code.

**Problem 3.** [70 pts]

In this Problem, you have to write a programmable Postfix calculator. The calculator will have an assign function which takes a String assigning a one-character variable to a value, expressed as a valid postfix expression. The calculator will also have an evaluate function which takes a postfix expression and returns the integer value that the expression evaluates to. Postfix expressions will consist of single character variables, arithmetic operations (+, -, *, /), and integers between 0 and 9 inclusive. You can see example use of the calculator below.

```java
Calculator c = new Calculator();
c.assign("x=5");
c.assign("y=x7+");
int first = c.evaluate("xy*");
c.assign("z=57*");
c.assign("x=x8+y*"());
int sec = c.evaluate("xy+");
int third = c.evaluate("z");
//This code will set first to 60, then it will set sec to 168, and third to 35.
```

You need to implement the following:
• Linked list (10 Points). You will create a linked list data structure to store variables. The linked list will consist of nodes where the elements are Variables. A Variable is a class you should create that has two members: its name, which is a char, and its value, which is an int. The linked list of Variables should be defined a separate class named VList.

• Iterator (10 points). VList should implement the Iterable interface. You should use your Iterator when looking up the values of variables.

• Stack (10 Points). You will create a stack to evaluate postfix expressions. You should only need one stack to perform each evaluation. The stack should also be its own class named CStack.

• Calculator (30 Points). Finally, you will create a calculator class named Calculator which contains a constructor and two public methods void assign(String statement) and int evaluate(String expr) as described above.

• Handling bad input (10 points). If your calculator encounters a malformed postfix expression or an undefined variable, it must throw a malformedPostfixException or undefinedVariableException, respectively. To define your own exceptions you need to extend the Exception class.

Submission. Please submit all files necessary for compilation, including the files named above and any additional files that you used. You do not need to submit a test file or a main method.