CS2121/9643 – Assignment 2
due Mar. 8, 2016 (latest to submit: Mar. 11)

1. (20pt) C++ contains the *conditional operator* "`?:`" defined as follows:

   ```
   boolean_expression ? value_if_true : value_if_false
   ```

   The value of the expression is `value_if_true` if `boolean_expression` is true and `value_if_false` otherwise. For example, maximum between `a` and `b` can be computed as:

   ```
   max = (a > b) ? a : b
   ```

   Translate the following expressions from infix into postfix:

   - `(((a + b) * (c + d) + a) * c - 6) * b`
   - `y += x + (-3) * z`
   - `max = (a > b) ? a : b`

   (The `'-'` in `(-3)` is unary, `'+='` is a single operator of arity 2, and `'?:'` is a single operator of arity 3.)

2. (20pt) Explain how to implement two stacks in a single array with n elements in such a way that neither stack overflows unless the total number of elements in both stacks together is n. The `push(item)` and `pop()` operations should still run in time $\mathcal{O}(1)$. Explain why they do.

3. (20pt) Show how to implement a queue using two stacks. Analyze the running time of the queue operations `enqueue(item)` and `dequeue()`. Explain your analysis.

4. (40pt) (Problem 7.2.(b) in textbook) Write a Python program `checkParentheses.py` to check whether a C++ source code has correctly matched parentheses: `(,[,{` with `),],}`, respectively. The parentheses within literal strings (enclosed between double quotes: `"parentheses (, [, {, ), ], }, within here should be ignored"`) and literal characters (enclosed between single quotes: `'['`) have to be ignored. For instance, the C++ code

   ```
   int main ()
   {
     char s[] = "Parentheses types: ([{";
     char c = ')';
     cout << s[10] << c;
     cout << (2+4)*9;
     return 0;
   }
   ```

   is correct. You only have to expand the code of the `isValidSource()` function on page 202 of textbook (download `ch07_code.zip`) to ignore parentheses within strings and characters.

   In addition, in case of an unmatched parenthesis, your program should output also the unmatched parenthesis and its line in the source code (as you would expect from a compiler). For example, for the C++ code below

   ```
   int sumList( int theList[], int size )
       {
         int sum = 0;
         int i = 0;
         while( i < size ) {
           sum += theList[ i ];
           i += 1;
         )
         return sum;
       }
   ```

the program will output

```
unmatched ) in line 8
```

while for the program

```
int sumList( int theList[], int size )
    {
      int sum = 0;
      int i = 0;
      while( i < size ) {
        sum += theList[ i ];
        i += 1;
      }
      return sum;
```

the program will output

```
unmatched { in line 2.
```

The program will read the C++ source code from a file given in the command line. You can either type in a terminal

```
python checkParentheses.py <c_code.cpp>
```

or, in Canopy,

```
run checkParentheses.py <c_code.cpp>
```

The input C++ code will not contain comments (so that you don't have to bother ignoring parentheses within those).

The structure of the code will not be considered for grading but it is strongly advised to implement clear logic, use meaningful names, and provide useful comments.

5. (bonus: 10pt) Give an algorithm that reverses a singly linked list of n elements, running in time $\mathcal{O}(n)$ and using $\mathcal{O}(1)$ additional space. Explain the algorithm and the time and space analysis.

Note: Submit your solution on `owl.uwo.ca`: .py files for question 4 and a .pdf file with all the remaining answers. Do not submit Python code for questions other than 4! Write your answers in pseudocode (preferred) or plain English.