COMP 352 PROGRAMMING ASSIGNMENT 2
ARASH SHAFIEE (40278142)

Pseudo code of parenthesis comparison

```
Class EqualParenthesis:

Method Validatefile(String filename)
    As long as there are more lines in the file it
    readss the current line, calls `isValid(line)` to
    check if the line has balanced parentheses, print
    the result indicating whether it is valid, and then
    increment `lineCounter` by 1.


Method isValid(s: String)
    Initialize openStack and closeStack as empty stacks

    For each character c in string s
        If c is '('
            Push '(' onto openStack
        Else If c is ')'
            If openStack is not empty
                Pop the top element from openStack
            Else If closeStack is not empty
                Pop the top element from closeStack
            Else
                Return false
        Else If c is '*'
            Push '*' onto closeStack
        Else If c is '$'
            Break the loop

    While both openStack and closeStack are not empty
        Pop one element from openStack
        Pop one element from closeStack

    Return true if openStack is empty, otherwise return
false
```

Class stack:

Method isEmpty()
Return true if top is -1, otherwise return false

Method isFull()
Return true if top is equal to capacity - 1, otherwise return false

Method expandStack()
Set newCapacity to capacity * 2 Create newStack as an array of size newCapacity Copy elements from stack to newStack up to index top Set stack to newStack Update capacity to newCapacity Print "Stack is full and will expand to size: " + newCapacity

```
Method push (c: char)
    If stack is full
        Call expandStack()

    Increment top by 1
    Set stack[top] to c

Method pop ()
    If stack is empty
        Print "Stack is empty"
        Return '\0' (null character as an error
code)

    Return stack[top] and decrement top by 1

Method top ()
    If stack is empty
        Print "Stack is empty"
        Return '\0' (null character as an error
code)
```

```
    Return stack[top]

Method size ()
    Return top + 1

Class Main
     in the main class we are only going to create
     an object from EqualParanthesis and read the
     text source file using validateFile
```

Big O complexity

Name of the method	Time complexity	explanation
IsEmpty	O (1)	This method only compares top to -1, which is a constant-time operation.
isFull	O (1)	This method checks if top equals capacity - 1, which takes constant time.
ExpandStack	O (n)	When the stack is full, this method doubles its size. Copying all existing elements to a new array of double capacity requires O(n)O(n) time since it copies each element once.
Push	O (1)	adds an element to the stack in constant time. However, when the stack is full, expandStack is called, which takes O(n) time. Since doubling happens infrequently, the amortized complexity of push remains O(1).
Pop	O (1)	Removing and returning the top element requires decrementing top and accessing the array, both of which are constant-time operations.
Top	O (1)	This method checks and returns the top element without removing it, which is constant-time.
size	O (1)	This method returns the size of the stack by calculating top + 1, a constant-time operation.

Name of the method	space complexity	explanation
isEmpty	O (1)	use a constant amount of space
isFull	O (1)	use a constant amount of space
expandStack	O (n)	The constructor allocates an array with an initial capacity of 2. This space is allocated in memory.
Push	O (1)	use a constant amount of space
Pop	O (1)	use a constant amount of space
Top	O (1)	use a constant amount of space
size	O (1)	use a constant amount of space