

Name: _____ Net ID: _____ (ex: abc123)

CSCI-UA.0002 – Lists

1. One tedious aspect of programming is keeping track of opened and closed parentheses. Every open parentheses needs a corresponding closing parentheses. If the parentheses are all correctly opened and closed, then the parentheses are *balanced*. Write a function that determines whether or not a series of open and closed parentheses are balanced. DO NOT JUST COUNT the number of left and right parentheses and check if they're equal! Instead...
 1. Create a function called `balanced_parentheses`
 2. It should take one argument, a string of opened and closed parentheses
 3. It should return True or False depending on whether or not all opened parentheses have a corresponding close, and are in the *proper order*!
 4. AGAIN, DO NOT COUNT THE NUMBER OF OPEN AND CLOSED PARENTHESES AND COMPARE
 5. **Use the algorithm below:**

Start with an empty list to temporarily store open parentheses

Go through every character in the string

 If you see an open parentheses add it to the list

 But if you see a close parentheses, remove the last open from your list

 If there's no open to remove... uh-oh! Not balanced!

If there are still open parentheses at the end of your list, not balanced!

* hint: Nested if statements may be useful

* hint: The pop method may be useful (or slicing may work too)

Examples:

all return True

```
print(balanced_parentheses('()()()()'))  
print(balanced_parentheses('(((())')))  
print(balanced_parentheses('()((())()')))
```

all return False

```
print(balanced_parentheses('()')  
print(balanced_parentheses('((((())'))  
print(balanced_parentheses('()'))  
print(balanced_parentheses('((((())'))  
print(balanced_parentheses('()()')  
print(balanced_parentheses('()()'))
```