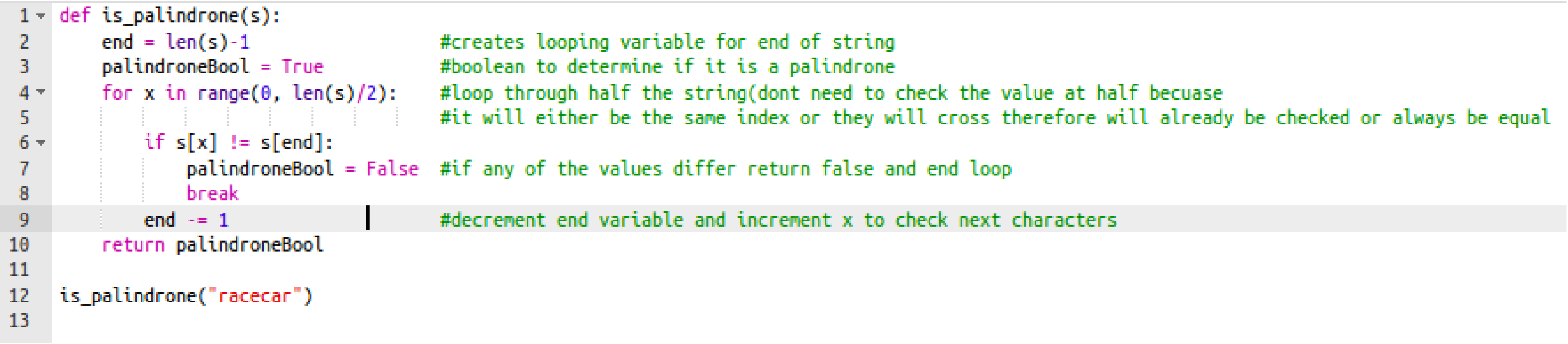
Johnny Persichilli
Phillies Questionnaire

```
def is_palindrone(s):
    r=""                    // this creates an empty string that the concatinated
                            letters will be added to
    for c in s:             //for each that iterates through string
        r = c +r            //concatenates letters to r backwards
    for x in range(0, len(s)):          // gets all indexes of string
        if s[x] == r[x]:                // if string = concatenated string for all
                                        //chars return true
            x = True                    //or else return false
        else:
            return False
    return x
```

This function runs on a time complexity of O(n^2) because the function must iterate first through the entire string, s, and assign it in reverse order to string r. Next, the function must iterate through every index or character of the two strings again to see if they are equal creating a function with a O(n^2) complexity. This function can be simplified using the definition of a palindrome, a word that is the same forwards as it is backwards, to O(n) complexity by having a two control variables, one that starts at the beginning of the string(index 0) and another that starts at the end of the string(index len(s)). These control variables will loop until they meet and be compared at each index to determine whether or not they are equal at each index. This only requires len(s)/2 iterations or a time complexity of O(n). The following code exemplifies that:



```
def is_palindrone(s):
    end = len(s)-1                  #creates looping variable for end of string
    palindroneBool = True           #boolean to determine if it is a palindrone
    for x in range(0, len(s)/2):    #loop through half the string(dont need to check the value at half becuase
                                    #it will either be the same index or they will cross therefore will already be checked or always be equal
        if s[x] != s[end]:
            palindroneBool = False  #if any of the values differ return false and end loop
            break
        end -= 1                    #decrement end variable and increment x to check next characters
    return palindroneBool

is_palindrone("racecar")
```