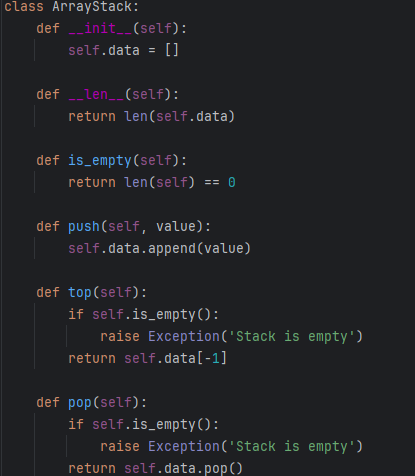
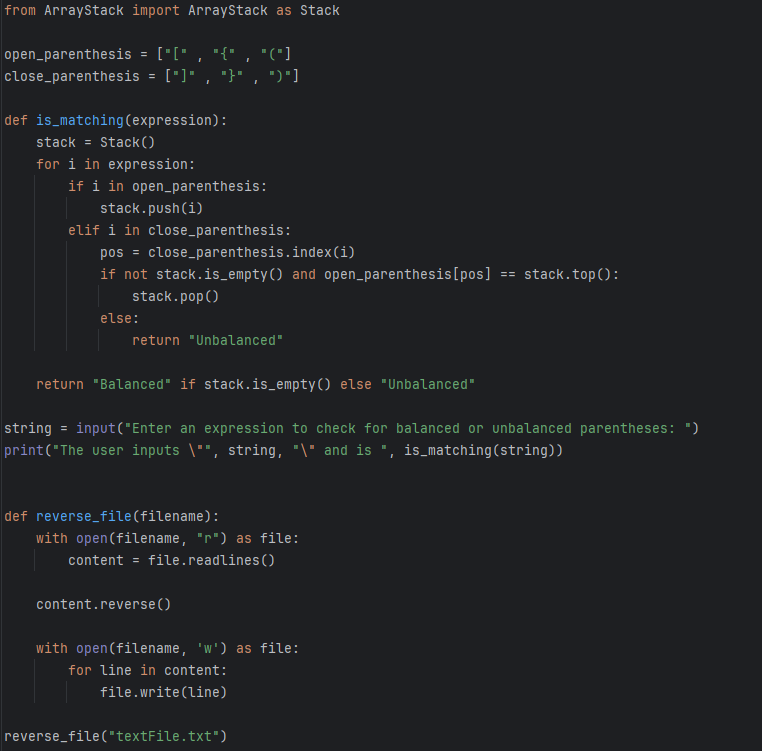
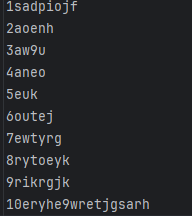
Charles Hanzel D. Gerardo  
DSALGO  
IDB2  
October 18, 2024

Code (ArrayStack)  
  
class ArrayStack:  
 def \_\_init\_\_(self):  
 self.data = []  
  
 def \_\_len\_\_(self):  
 return len(self.data)  
  
 def is\_empty(self):  
 return len(self) == 0  
  
 def push(self, value):  
 self.data.append(value)  
  
 def top(self):  
 if self.is\_empty():  
 raise Exception('Stack is empty')  
 return self.data[-1]  
  
 def pop(self):  
 if self.is\_empty():  
 raise Exception('Stack is empty')  
 return self.data.pop()

Code(main)  
  
from ArrayStack import ArrayStack as Stack  
  
open\_parenthesis = ["[" , "{" , "("]  
close\_parenthesis = ["]" , "}" , ")"]  
  
def is\_matching(expression):  
 stack = Stack()  
 for i in expression:  
 if i in open\_parenthesis:  
 stack.push(i)  
 elif i in close\_parenthesis:  
 pos = close\_parenthesis.index(i)  
 if not stack.is\_empty() and open\_parenthesis[pos] == stack.top():  
 stack.pop()  
 else:  
 return "Unbalanced"  
  
 return "Balanced" if stack.is\_empty() else "Unbalanced"  
  
string = input("Enter an expression to check for balanced or unbalanced parentheses: ")  
print("The user inputs \"", string, "\" and is ", is\_matching(string))  
  
  
def reverse\_file(filename):  
 with open(filename, "r") as file:  
 content = file.readlines()  
  
 content.reverse()  
  
 with open(filename, 'w') as file:  
 for line in content:  
 file.write(line)  
  
reverse\_file("textFile.txt")

Text file (textFile)  
  
1sadpiojf  
2aoenh  
3aw9u  
4aneo  
5euk  
6outej  
7ewtyrg  
8rytoeyk  
9rikrgjk  
10eryhe9wretjgsarh

Output  
