Lab task # 01:

Write a program to let the user enter a string of his own choice. Check whether the given string is a palindrome or not.

Code # 01:

```
string = input("Enter string: ")
string = string.lower()
stack = []

for letter in string:
  stack.append(letter)

print(stack)
reverse_stack = stack[::-1]
print(reverse_stack)

if stack == reverse_stack:
    print('The string is a palindrome.')
else:
    print('The string is not a palindrome.')
```

Proof :

#1 : Not a palindrome string

```
string = input("Enter the string : ")
string = string.lower()
stack = []

for letter in string:
  stack.append(letter)

print(stack)
reverse_stack = stack[::-1]
print(reverse_stack)

if stack == reverse_stack:
    print('A palindrome String')
else:
    print('Not a palindrome String')
```

```
Enter the string : king
['k', 'i', 'n', 'g']
['g', 'n', 'i', 'k']
Not a palindrome String
```

#2 : A palindrome string

```
string = input("Enter the string : ")
string = string.lower()
stack = []

for letter in string:
  stack.append(letter)

print(stack)
reverse_stack = stack[::-1]
print(reverse_stack)

if stack == reverse_stack:
    print('A palindrome String')
else:
    print('Not a palindrome String')
```

```
Enter the string : RACECAR
['r', 'a', 'c', 'e', 'c', 'a', 'r']
['r', 'a', 'c', 'e', 'c', 'a', 'r']
A palindrome String
```

Lab Task # 02:

Write a program to check the balanced parenthesis in the expression or not using stack

Code # 02:

```
def isBalanced(final_str):
    type_brackets = ['()', '{}', '[]']
    while any(x in final_str for x in type_brackets):
        for br in type_brackets:
            final_str = final_str.replace(br, '')
    return not final_str


string = "({[{()}]})"
print(string, "-", "Balanced"
      if isBalanced(string) else "Unbalanced")
```

Proof :

```
def isBalanced(final_str):
    type_brackets = ['()', '{}', '[]']
    while any(x in final_str for x in type_brackets):
        for br in type_brackets:
            final_str = final_str.replace(br, '')
    return not final_str


string = "({[{()}]})"
print(string, "-", "Balanced"
      if isBalanced(string) else "Unbalanced")
```

```
({[{()}]}) - Balanced
```

“THE END”