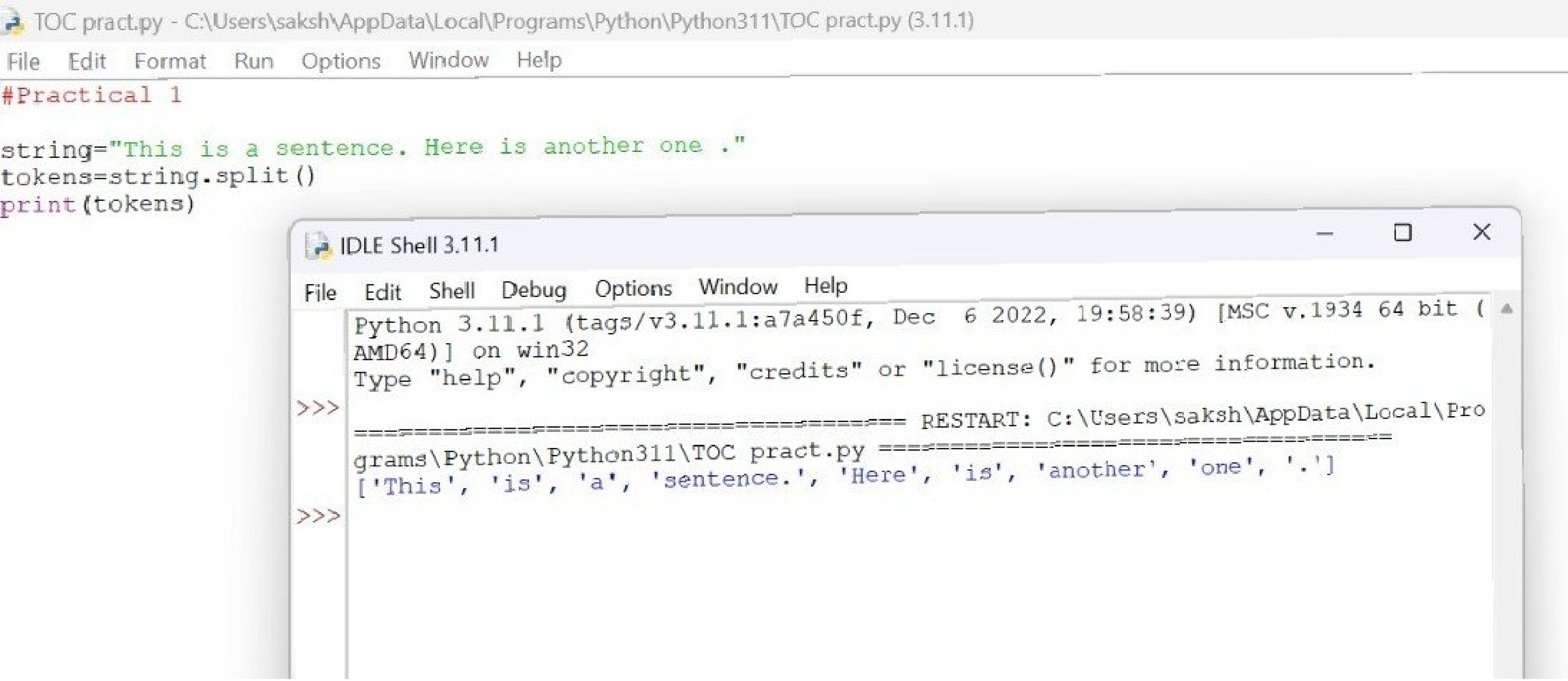
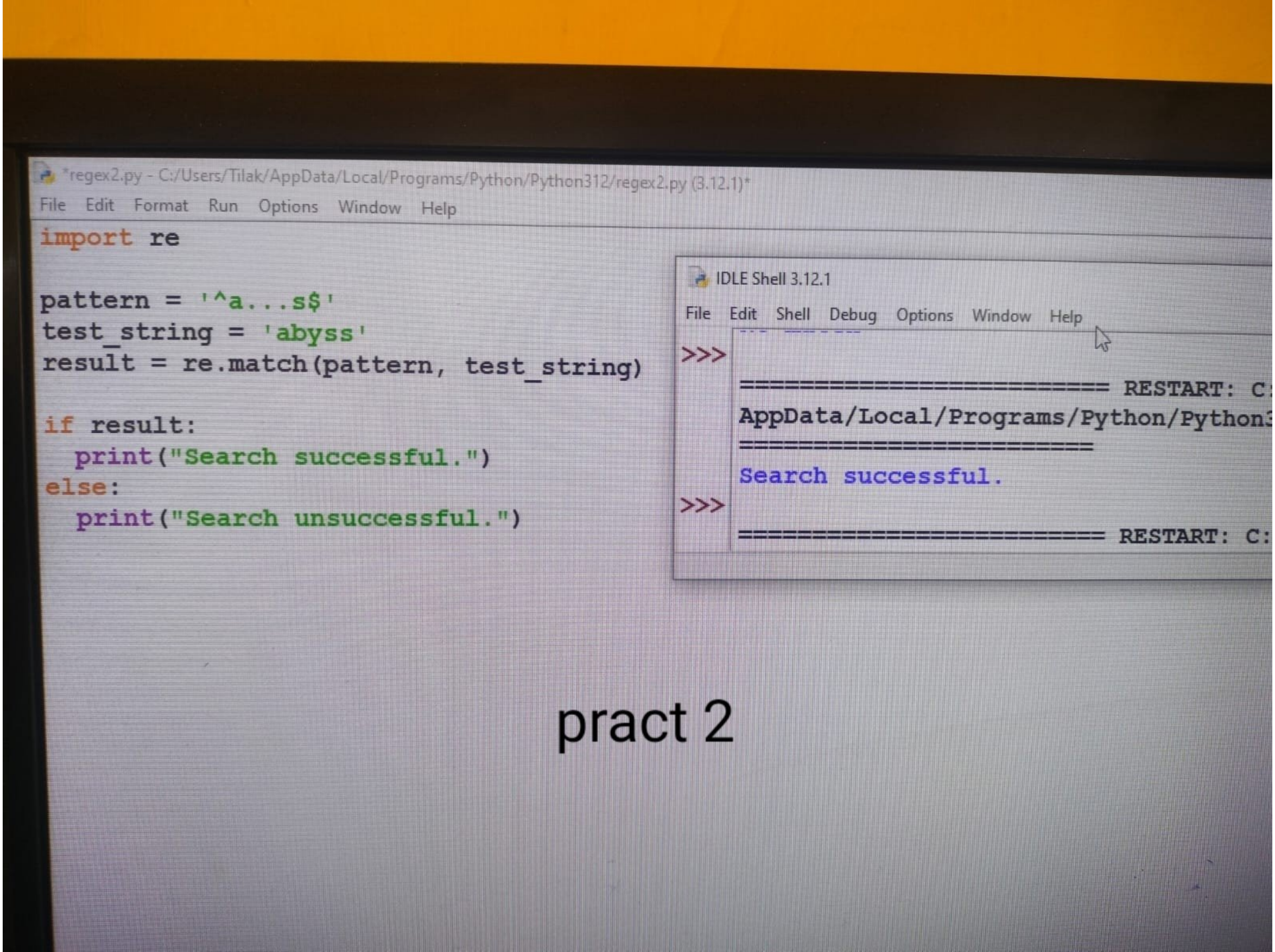

TOC pract.py - C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py (3.11.1)
File
Edit
Format
Run
Options
Window
Help
IDLE Shell 3.11.1
File
Edit
Shell
Debug
Options
Window
Help

```
#Practical 1

string="This is a sentence. Here is another one ."
tokens=string.split()
print(tokens)
```

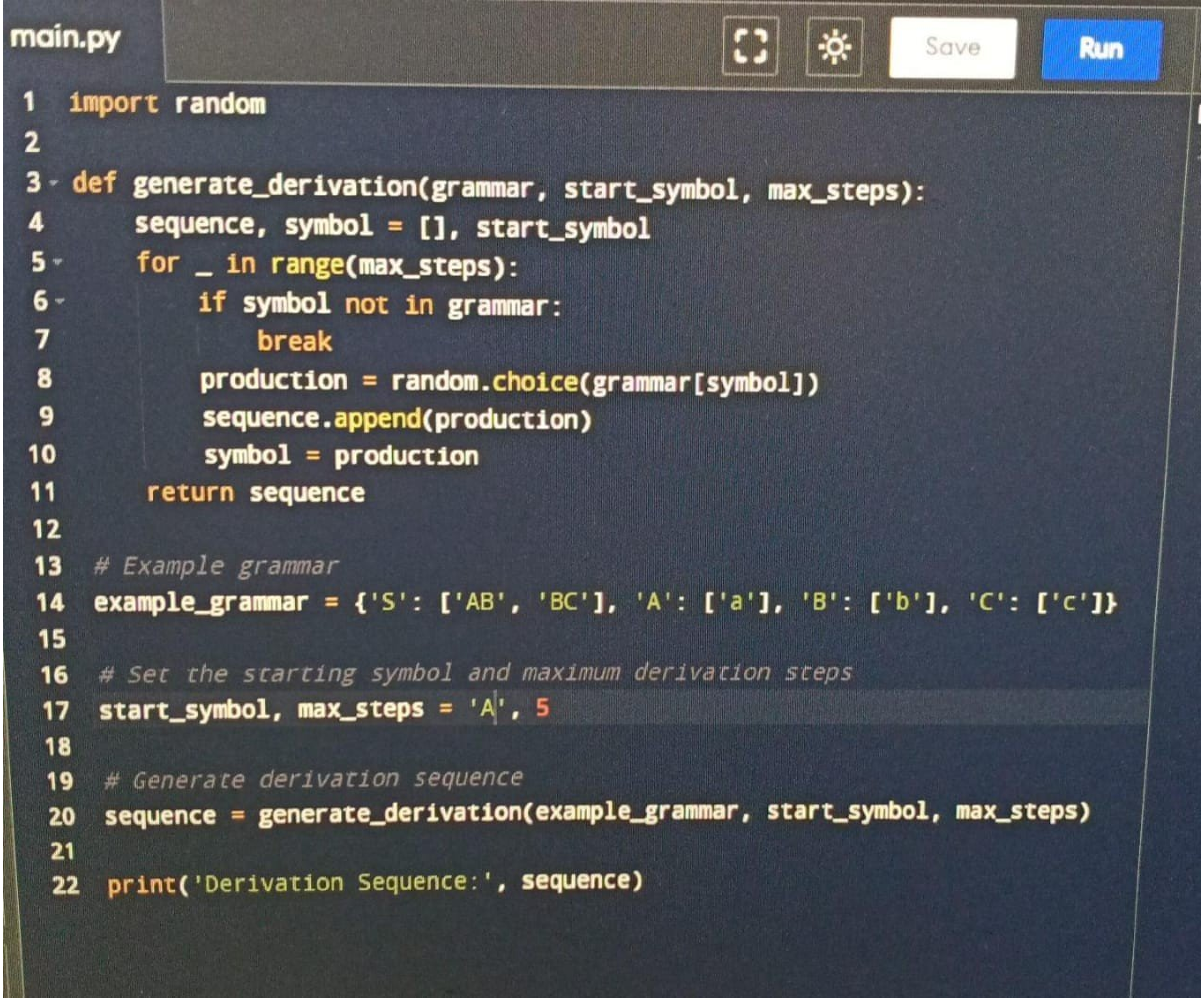
```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==================================== RESTART: C:\Users\saksh\AppData\Local\Pro
grams\Python\Python311\TOC pract.py ===================================
['This', 'is', 'a', 'sentence.', 'Here', 'is', 'another', 'one', '.']
>>> 
```


regex2.py - C:/Users/Tilak/AppData/Local/Programs/Python/Python312/regex2.py (3.12.1)
File Edit Format Run Options Window Help
IDLE Shell 3.12.1
File Edit Shell Debug Options Window Help

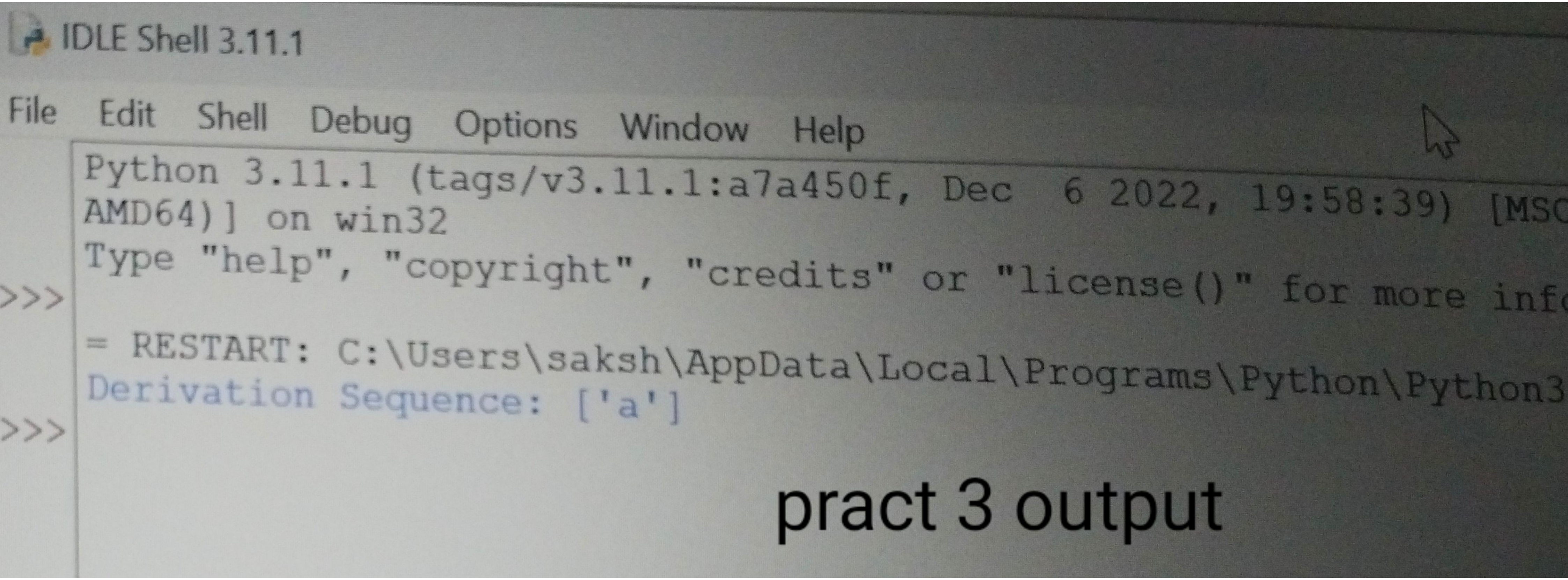
```python
import re

pattern = '^a...s$'
test_string = 'abyss'
result = re.match(pattern, test_string)

if result:
  print("Search successful.")
else:
  print("Search unsuccessful.")
```

```
>>>
    ========================= RESTART: C:
    AppData/Local/Programs/Python/Python3
    ========================
    Search successful.
>>>
    ========================= RESTART: C:
```

pract 2

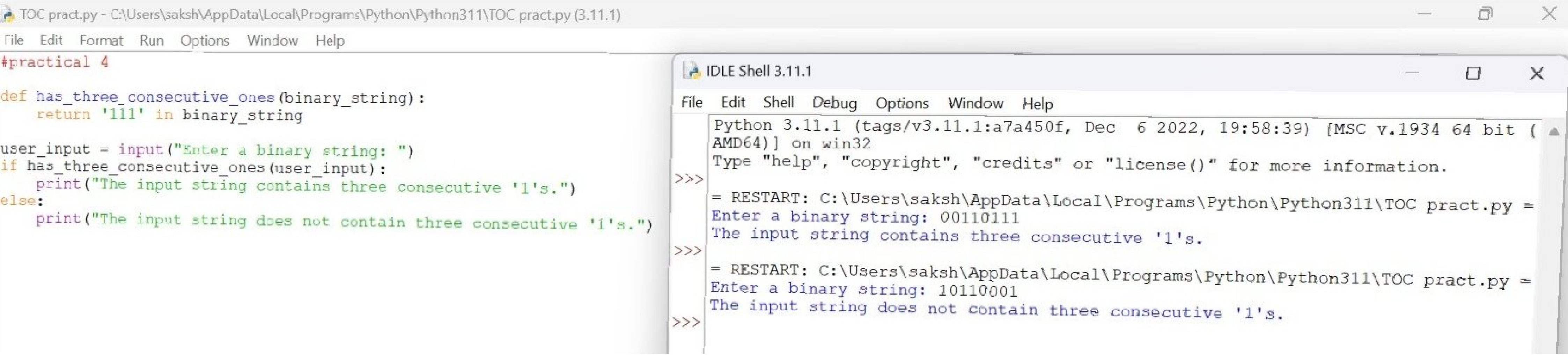


```python
import random

def generate_derivation(grammar, start_symbol, max_steps):
    sequence, symbol = [], start_symbol
    for _ in range(max_steps):
        if symbol not in grammar:
            break
        production = random.choice(grammar[symbol])
        sequence.append(production)
        symbol = production
    return sequence

# Example grammar
example_grammar = {'S': ['AB', 'BC'], 'A': ['a'], 'B': ['b'], 'C': ['c']}

# Set the starting symbol and maximum derivation steps
start_symbol, max_steps = 'A', 5

# Generate derivation sequence
sequence = generate_derivation(example_grammar, start_symbol, max_steps)

print('Derivation Sequence:', sequence)
```



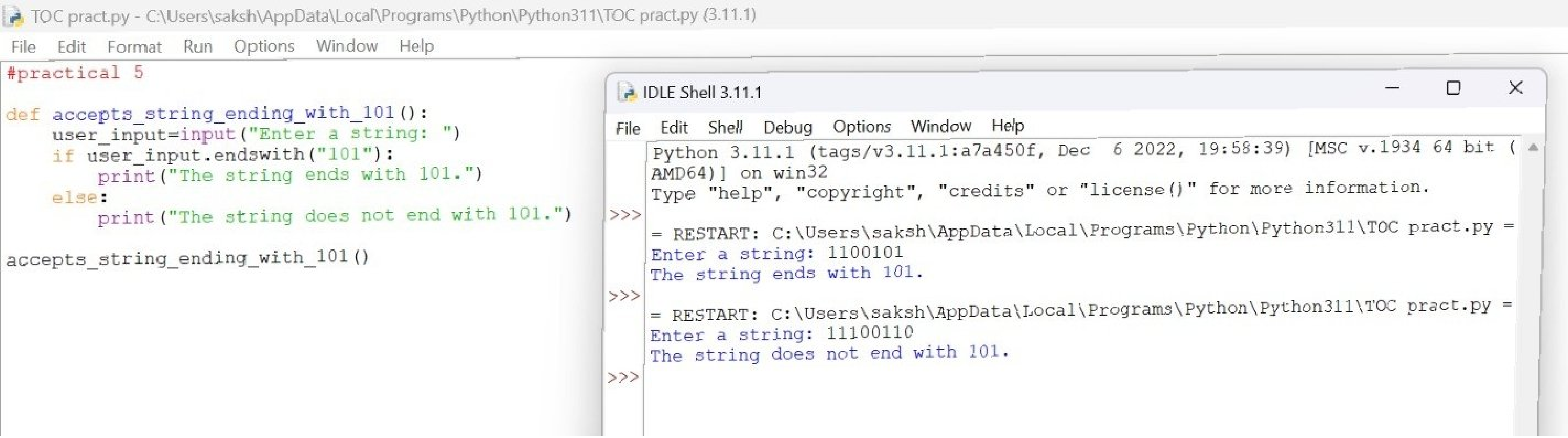
```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more info
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python3
Derivation Sequence: ['a']
>>>
```

pract 3 output


TOC pract.py - C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py (3.11.1)
File Edit Format Run Options Window Help
IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help

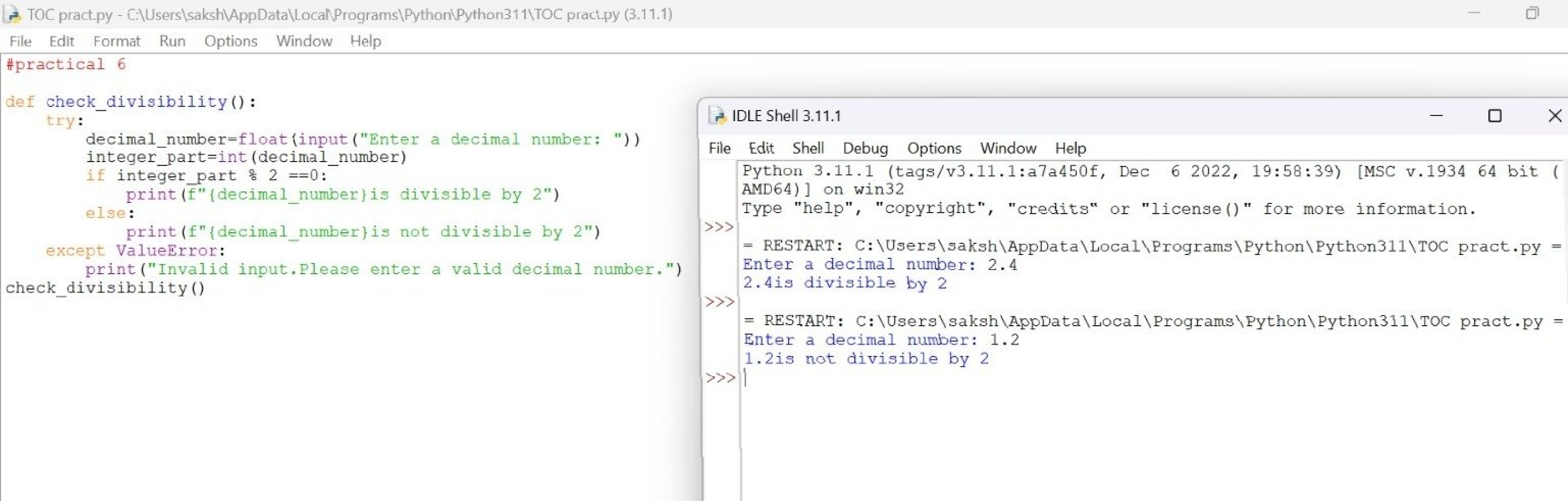
```python
#practical 4

def has_three_consecutive_ones(binary_string):
    return '111' in binary_string

user_input = input("Enter a binary string: ")
if has_three_consecutive_ones(user_input):
    print("The input string contains three consecutive '1's.")
else:
    print("The input string does not contain three consecutive '1's.")
```

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py =
Enter a binary string: 00110111
The input string contains three consecutive '1's.
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py =
Enter a binary string: 10110001
The input string does not contain three consecutive '1's.
>>>
```


TOC pract.py - C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py (3.11.1)
File Edit Format Run Options Window Help
IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help

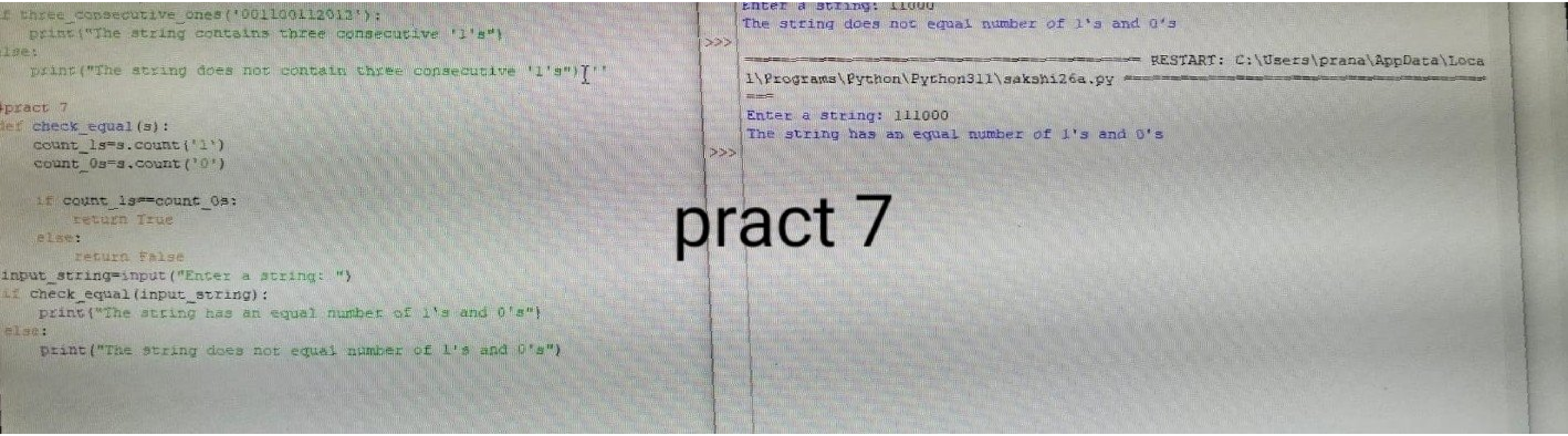
```python
#practical 5

def accepts_string_ending_with_101():
    user_input=input("Enter a string: ")
    if user_input.endswith("101"):
        print("The string ends with 101.")
    else:
        print("The string does not end with 101.")

accepts_string_ending_with_101()
```

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py =
Enter a string: 1100101
The string ends with 101.
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py =
Enter a string: 11100110
The string does not end with 101.
>>>
```


TOC pract.py - C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py (3.11.1)
File Edit Format Run Options Window Help
IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help

```python
#practical 6

def check_divisibility():
    try:
        decimal_number=float(input("Enter a decimal number: "))
        integer_part=int(decimal_number)
        if integer_part % 2 ==0:
            print(f"{decimal_number}is divisible by 2")
        else:
            print(f"{decimal_number}is not divisible by 2")
    except ValueError:
        print("Invalid input.Please enter a valid decimal number.")
check_divisibility()
```

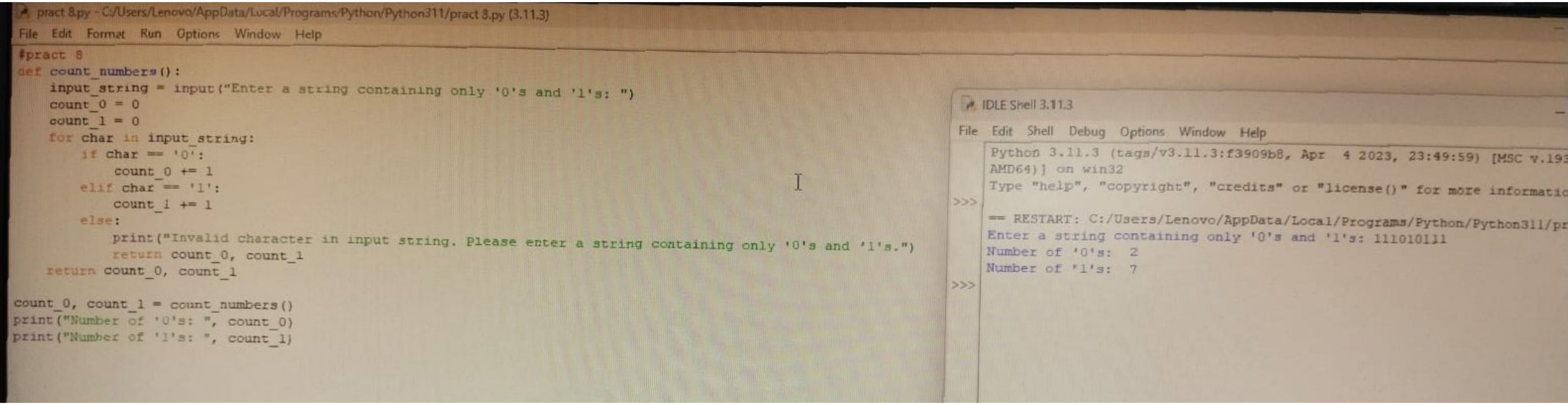
```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py =
Enter a decimal number: 2.4
2.4is divisible by 2
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py =
Enter a decimal number: 1.2
1.2is not divisible by 2
>>>
```



pract 7

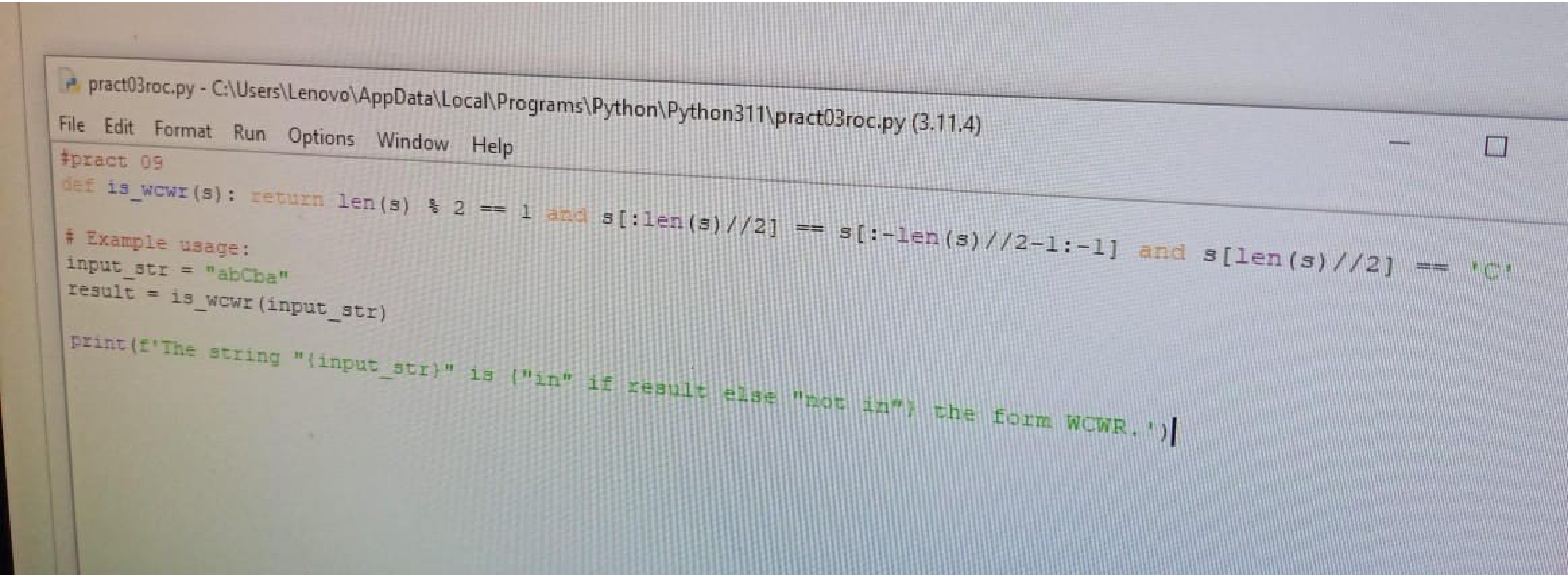
```
f three_consecutive_ones('001100112012'):
   print("The string contains three consecutive '1's")
lse:
   print("The string does not contain three consecutive '1's")''

pract 7
ef check_equal(s):
    count_1s=s.count('1')
    count_0s=s.count('0')

    if count_1s==count_0s:
        return True
    else:
        return False
input_string=input("Enter a string: ")
if check_equal(input_string):
    print("The string has an equal number of 1's and 0's")
else:
    print("The string does not equal number of 1's and 0's")
```

```
    Enter a string: 11000
    The string does not equal number of 1's and 0's
>>>
    ================================== RESTART: C:\Users\prana\AppData\Loca
    l\Programs\Python\Python311\sakshi26a.py ==================================
    ===
    Enter a string: 111000
    The string has an equal number of 1's and 0's
>>>
```


pract 8.py - C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/pract 8.py (3.11.3)
File Edit Format Run Options Window Help
IDLE Shell 3.11.3
File Edit Shell Debug Options Window Help

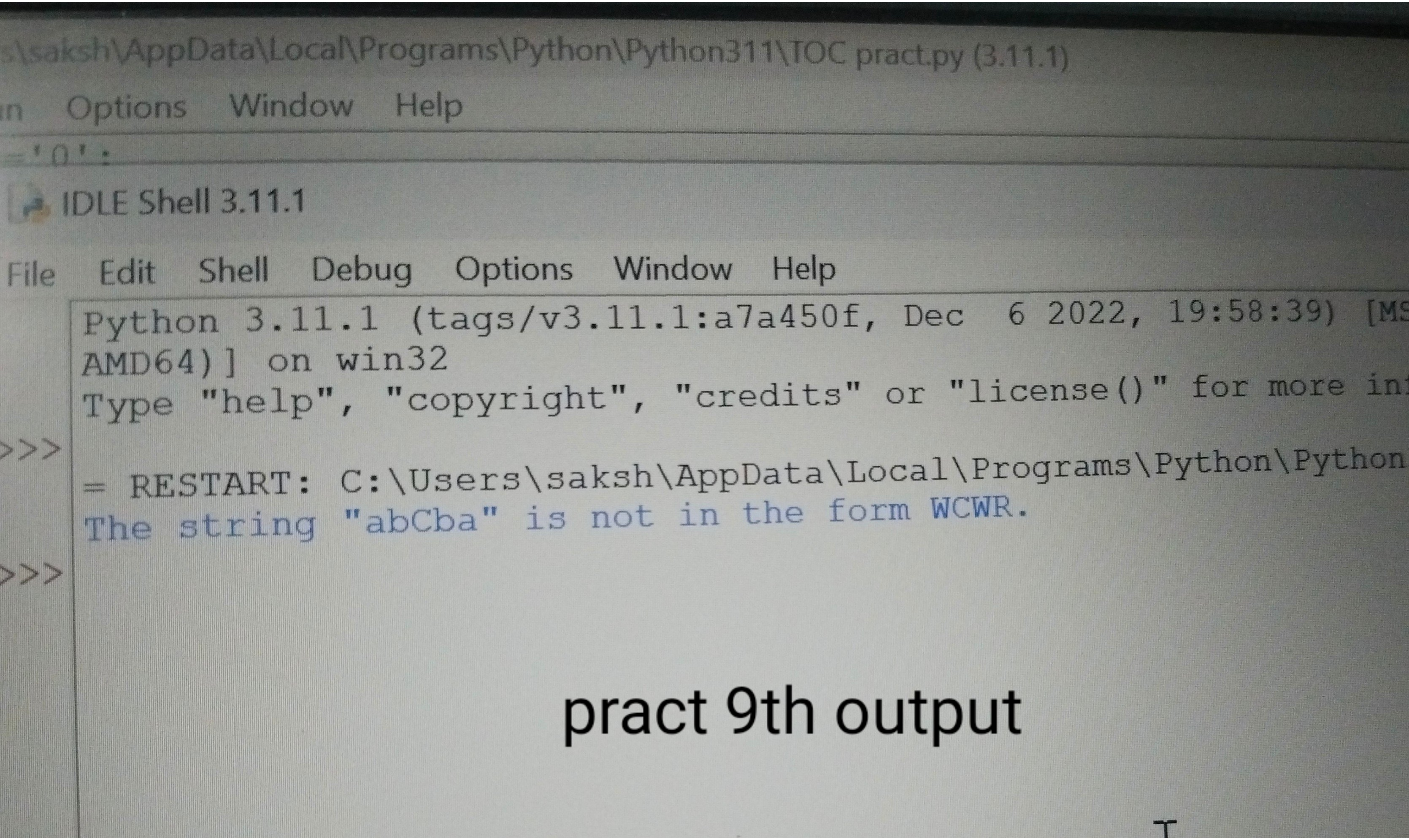
```python
#pract 8
def count_numbers():
    input_string = input("Enter a string containing only '0's and '1's: ")
    count_0 = 0
    count_1 = 0
    for char in input_string:
        if char == '0':
            count_0 += 1
        elif char == '1':
            count_1 += 1
        else:
            print("Invalid character in input string. Please enter a string containing only '0's and '1's.")
            return count_0, count_1
    return count_0, count_1

count_0, count_1 = count_numbers()
print("Number of '0's: ", count_0)
print("Number of '1's: ", count_1)
```

```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.193
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informatic
>>>
== RESTART: C:/Users/Lenovo/AppData/Local/Programs/Python/Python311/pr
Enter a string containing only '0's and '1's: 111010111
Number of '0's:  2
Number of '1's:  7
>>>
```



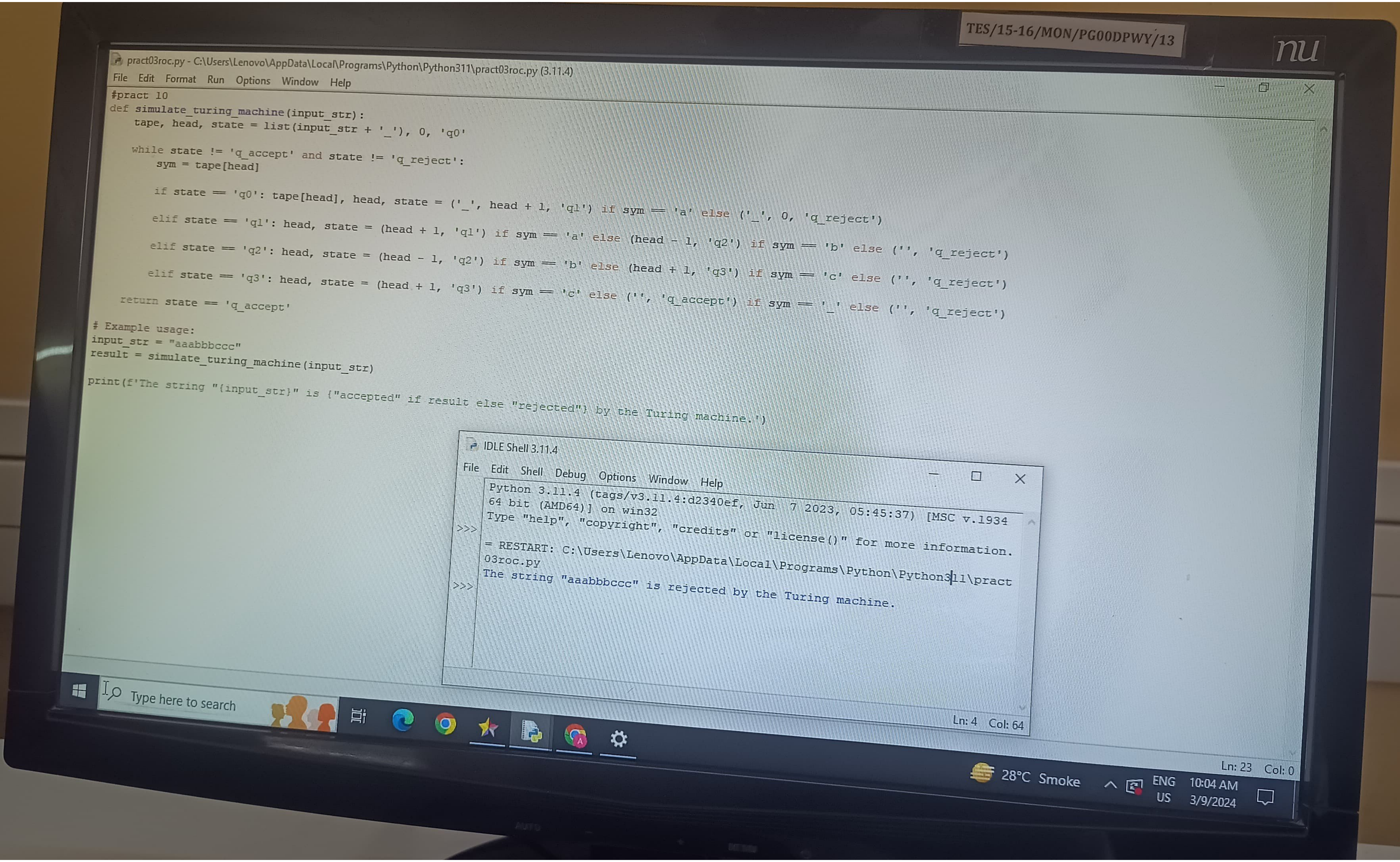
pract03roc.py - C:\Users\Lenovo\AppData\Local\Programs\Python\Python311\pract03roc.py (3.11.4)
File
Edit
Format
Run
Options
Window
Help

```
#pract 09
def is_wcwr(s): return len(s) % 2 == 1 and s[:len(s)//2] == s[:-len(s)//2-1:-1] and s[len(s)//2] == 'C'

# Example usage:
input_str = "abCba"
result = is_wcwr(input_str)

print(f'The string "{input_str}" is {"in" if result else "not in"} the form WCWR.')
```


s\saksh\AppData\Local\Programs\Python\Python311\TOC pract.py (3.11.1)
n Options Window Help
='0':
IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MS
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
>>>
= RESTART: C:\Users\saksh\AppData\Local\Programs\Python\Python
The string "abCba" is not in the form WCWR.
>>>
```

pract 9th output


TES/15-16/MON/PG00DPWY/13
nu
pract03roc.py - C:\Users\Lenovo\AppData\Local\Programs\Python\Python311\pract03roc.py (3.11.4)
File Edit Format Run Options Window Help
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Ln: 4 Col: 64
Ln: 23 Col: 0
Type here to search
28°C Smoke
ENG
US
10:04 AM
3/9/2024

```python
#pract 10
def simulate_turing_machine(input_str):
    tape, head, state = list(input_str + '_'), 0, 'q0'

    while state != 'q_accept' and state != 'q_reject':
        sym = tape[head]

        if state == 'q0': tape[head], head, state = ('_', head + 1, 'q1') if sym == 'a' else ('_', 0, 'q_reject')
        elif state == 'q1': head, state = (head + 1, 'q1') if sym == 'a' else (head - 1, 'q2') if sym == 'b' else ('', 'q_reject')
        elif state == 'q2': head, state = (head - 1, 'q2') if sym == 'b' else (head + 1, 'q3') if sym == 'c' else ('', 'q_reject')
        elif state == 'q3': head, state = (head + 1, 'q3') if sym == 'c' else ('', 'q_accept') if sym == '_' else ('', 'q_reject')
    return state == 'q_accept'

# Example usage:
input_str = "aaabbbccc"
result = simulate_turing_machine(input_str)

print(f'The string "{input_str}" is {"accepted" if result else "rejected"} by the Turing machine.')
```

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Lenovo\AppData\Local\Programs\Python\Python311\pract
03roc.py
The string "aaabbbccc" is rejected by the Turing machine.
>>>
```