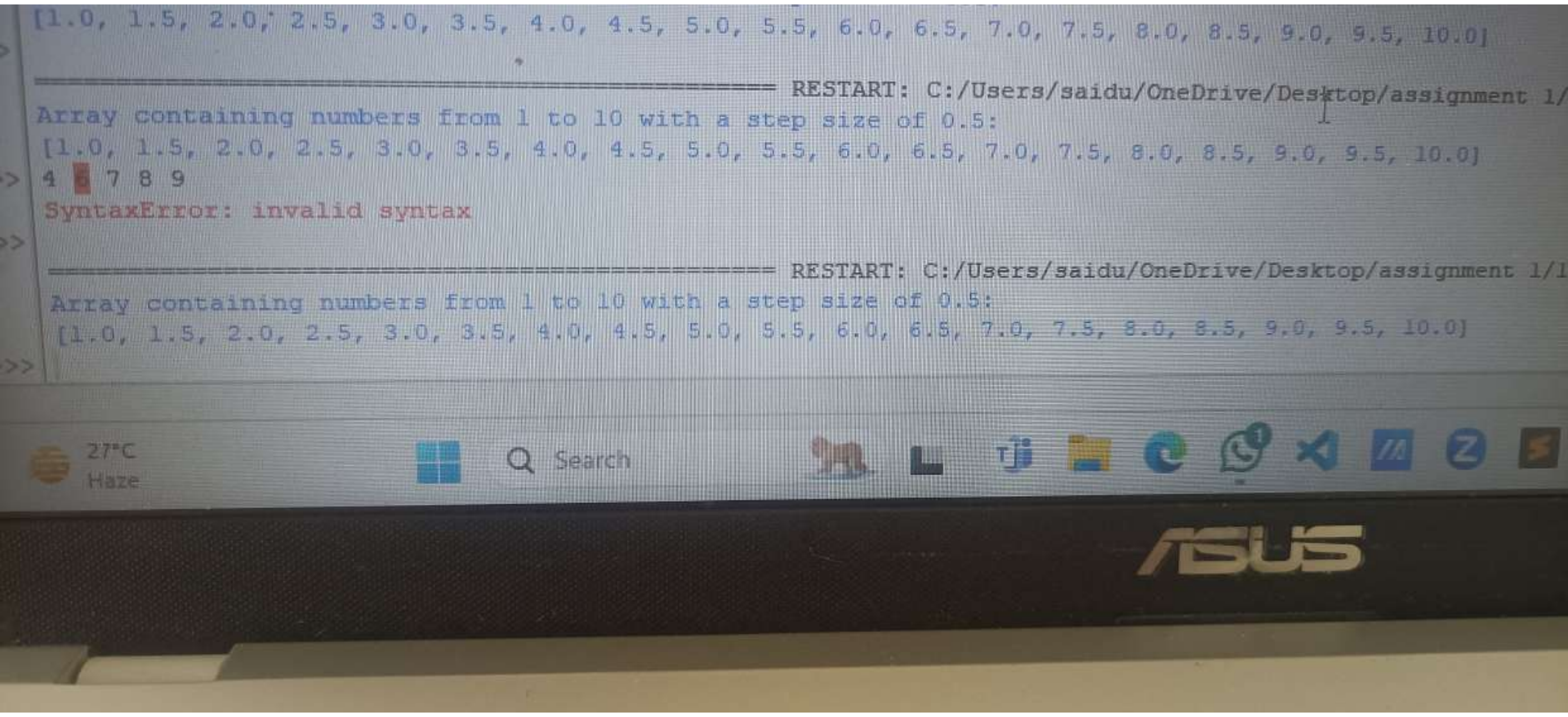
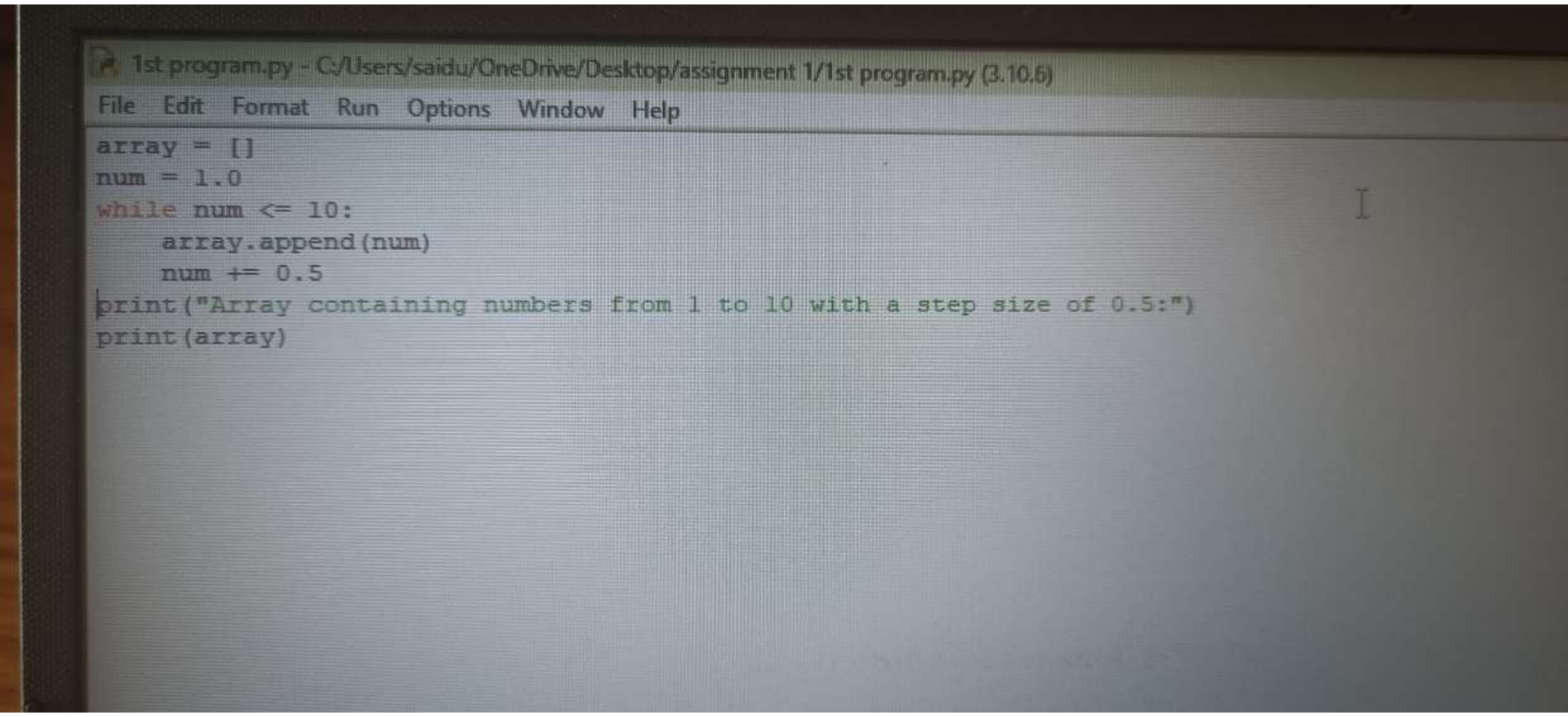
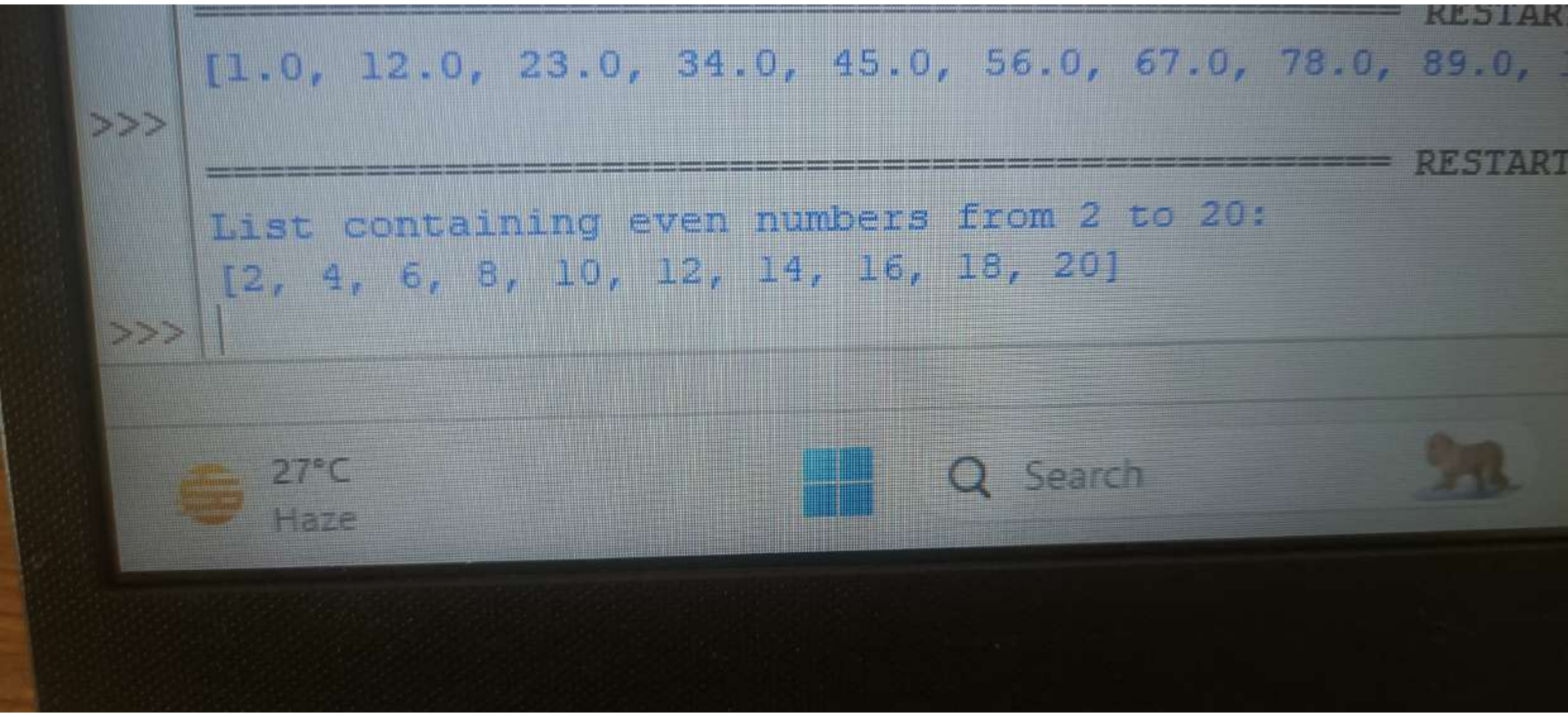


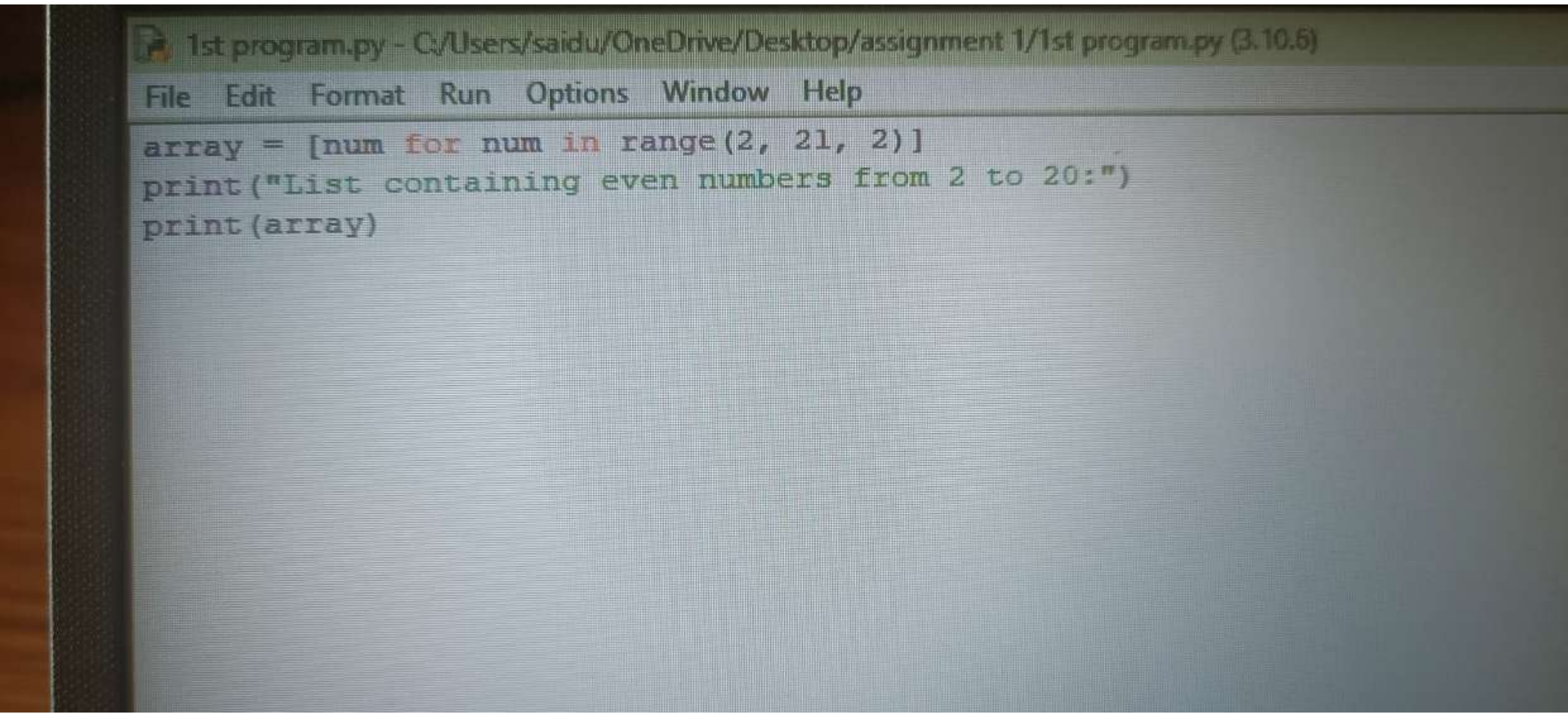
```
[1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0]
>
================================================= RESTART: C:/Users/saidu/OneDrive/Desktop/assignment 1/
Array containing numbers from 1 to 10 with a step size of 0.5:
[1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0]
>> 4 6 7 8 9
SyntaxError: invalid syntax
>>
================================================= RESTART: C:/Users/saidu/OneDrive/Desktop/assignment 1/1
Array containing numbers from 1 to 10 with a step size of 0.5:
[1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0]
>>
```


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File
Edit
Format
Run
Options
Window
Help

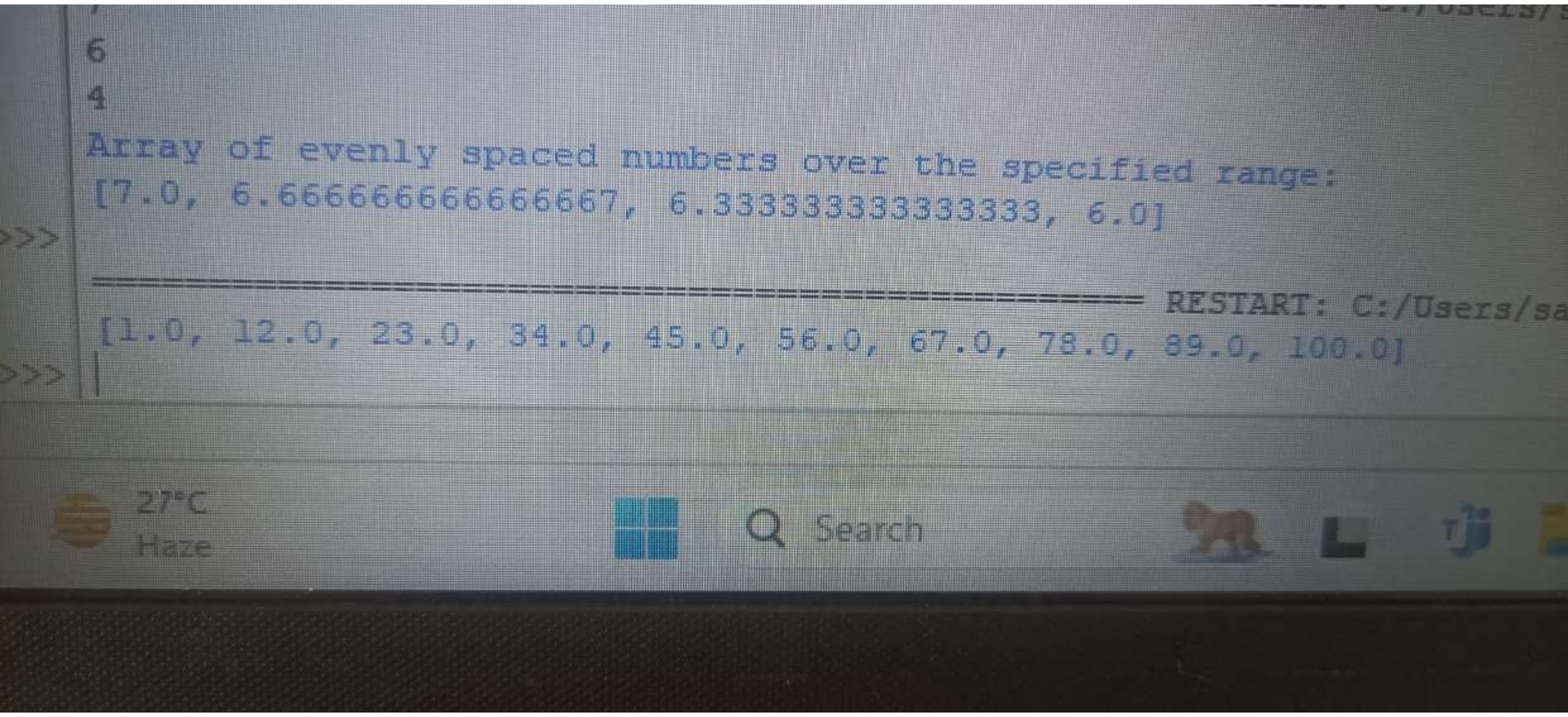
```
array = []
num = 1.0
while num <= 10:
    array.append(num)
    num += 0.5
print("Array containing numbers from 1 to 10 with a step size of 0.5:")
print(array)
```



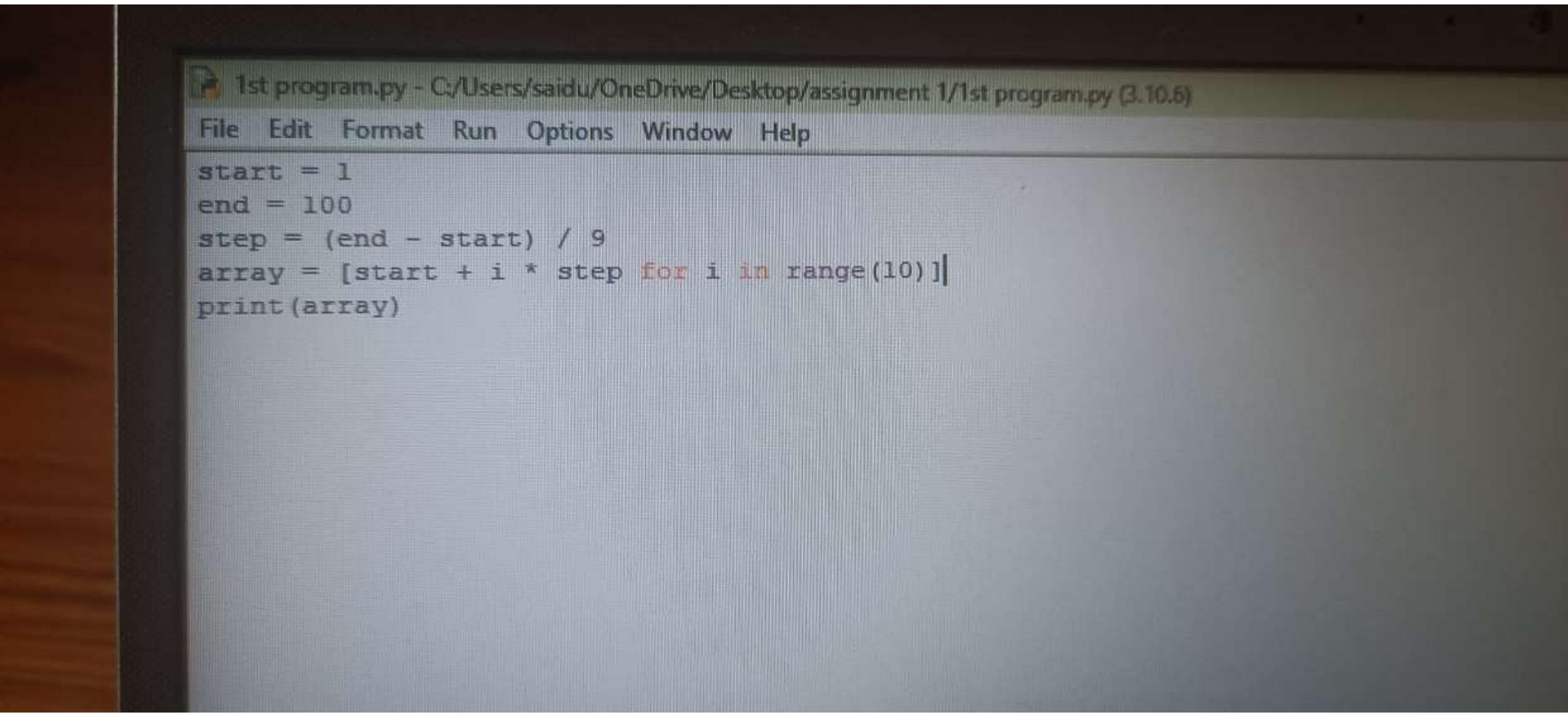
```
============================================================== RESTAR
[1.0, 12.0, 23.0, 34.0, 45.0, 56.0, 67.0, 78.0, 89.0,
>>>
============================================================= RESTART
List containing even numbers from 2 to 20:
 [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
>>> |
```


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File
Edit
Format
Run
Options
Window
Help

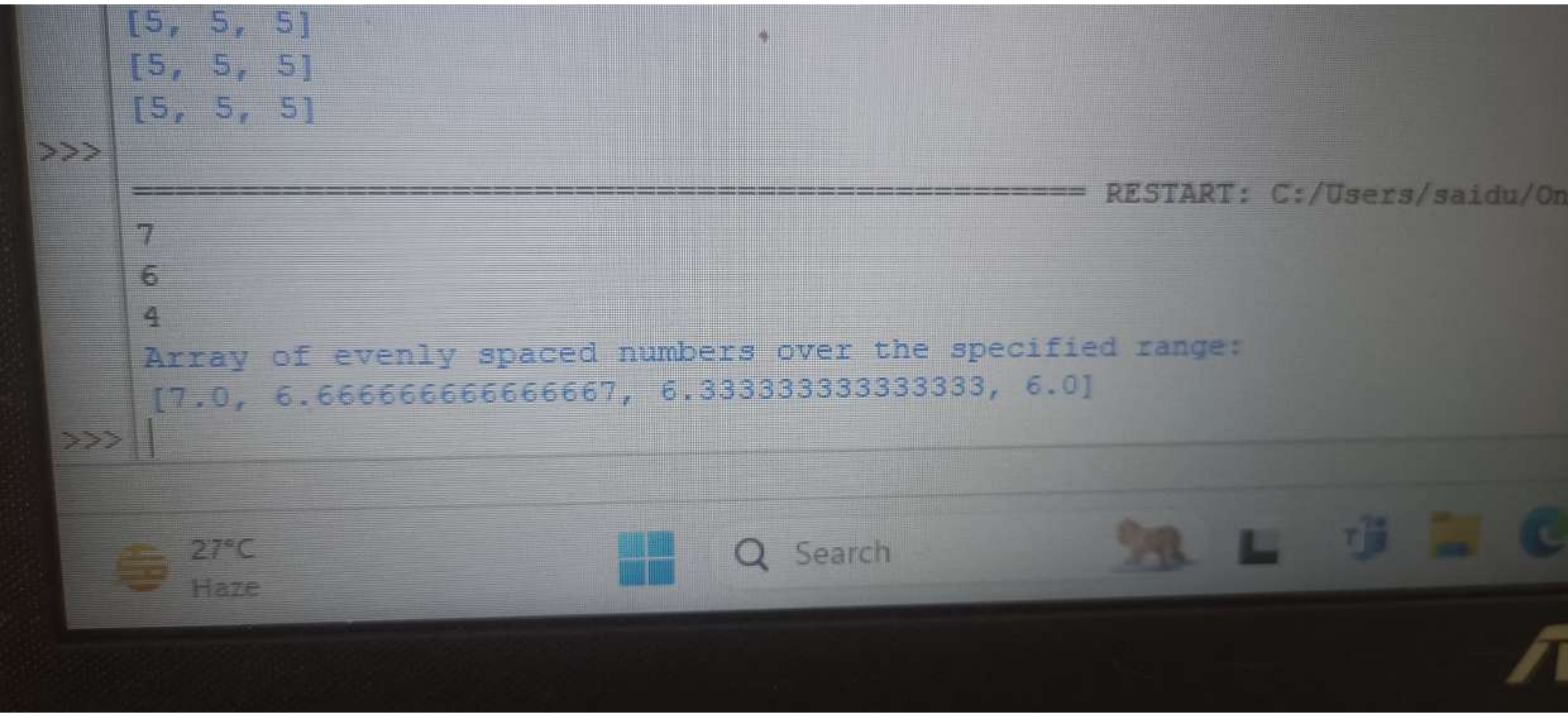
```
array = [num for num in range(2, 21, 2)]
print("List containing even numbers from 2 to 20:")
print(array)
```



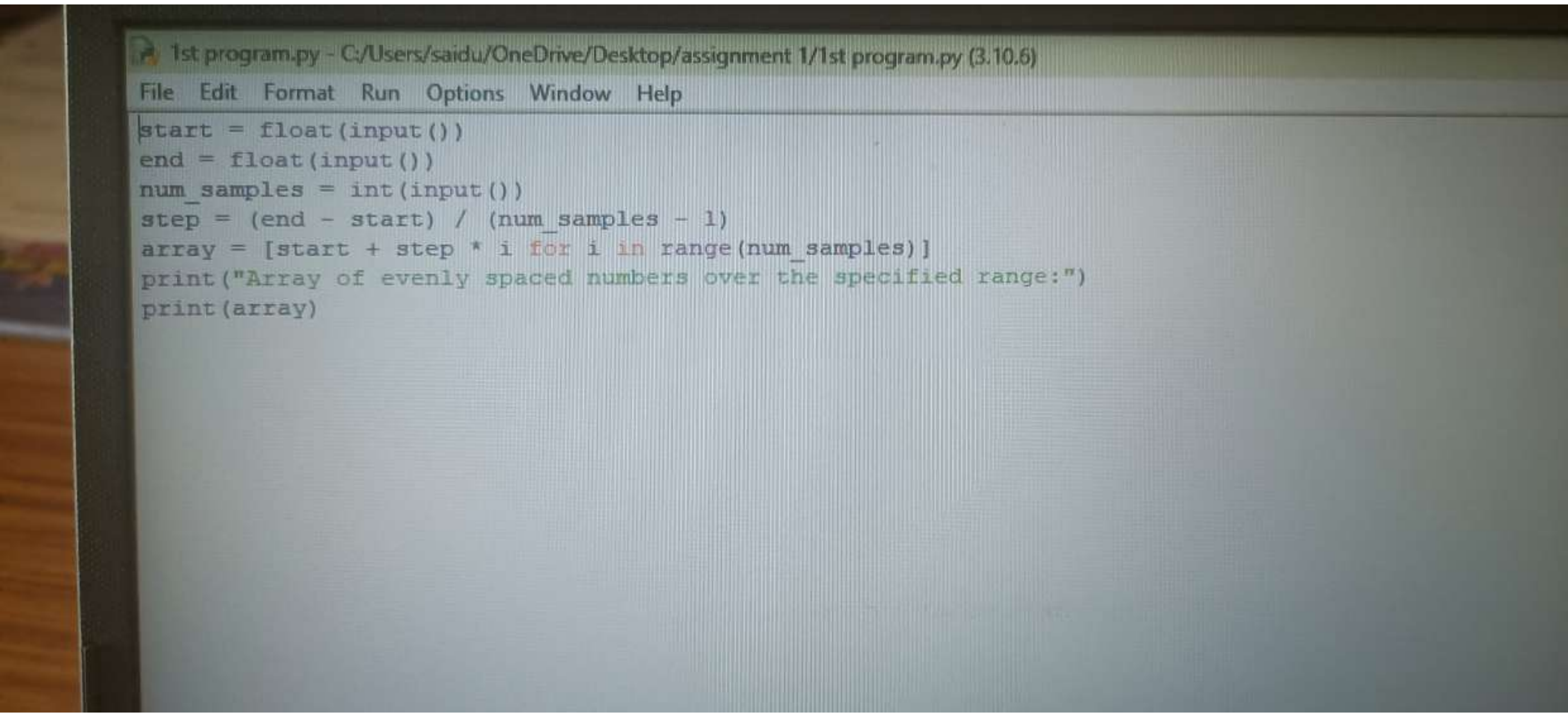
```
6
4
Array of evenly spaced numbers over the specified range:
[7.0, 6.666666666666667, 6.333333333333333, 6.0]
>>>
========================================== RESTART: C:/Users/sa
[1.0, 12.0, 23.0, 34.0, 45.0, 56.0, 67.0, 78.0, 89.0, 100.0]
>>> |
```


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

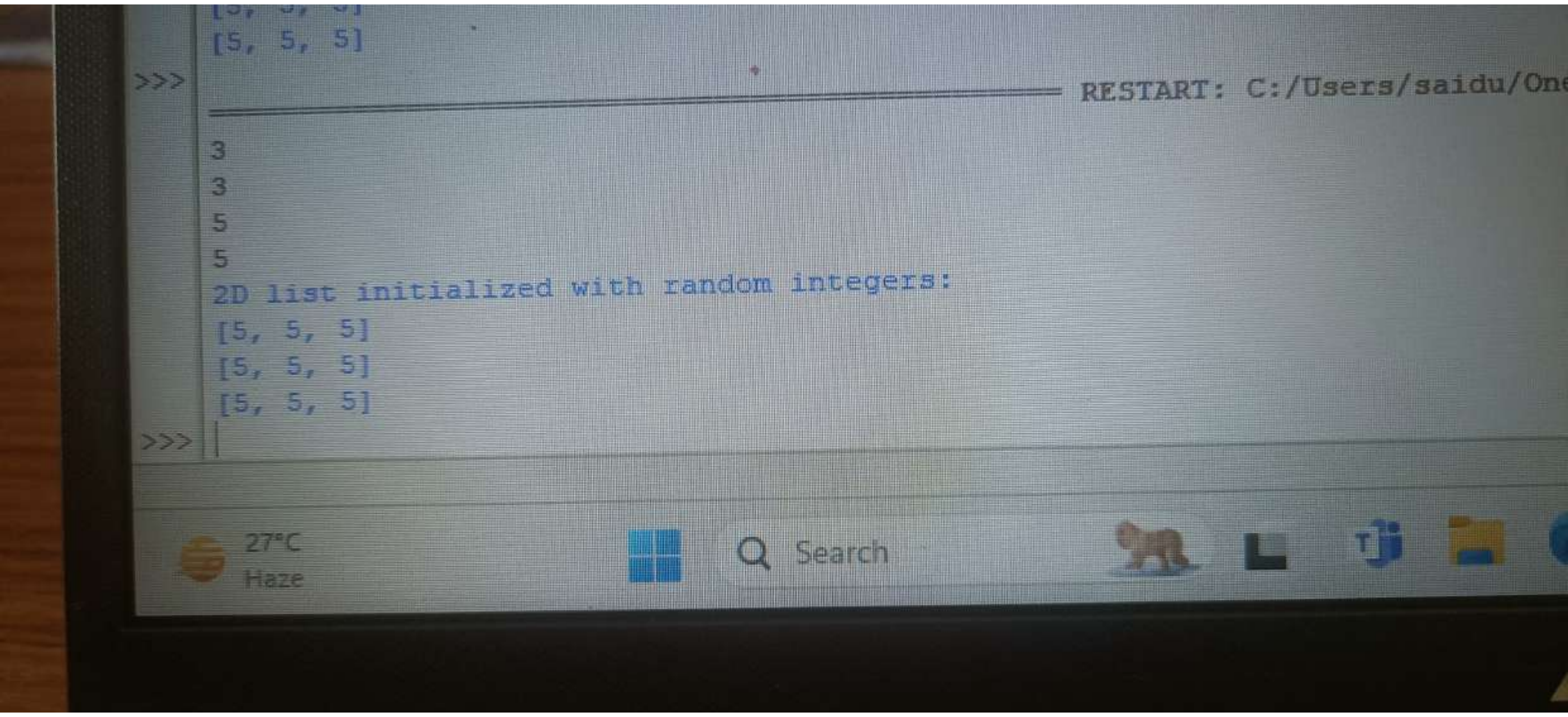
```
start = 1
end = 100
step = (end - start) / 9
array = [start + i * step for i in range(10)]
print(array)
```



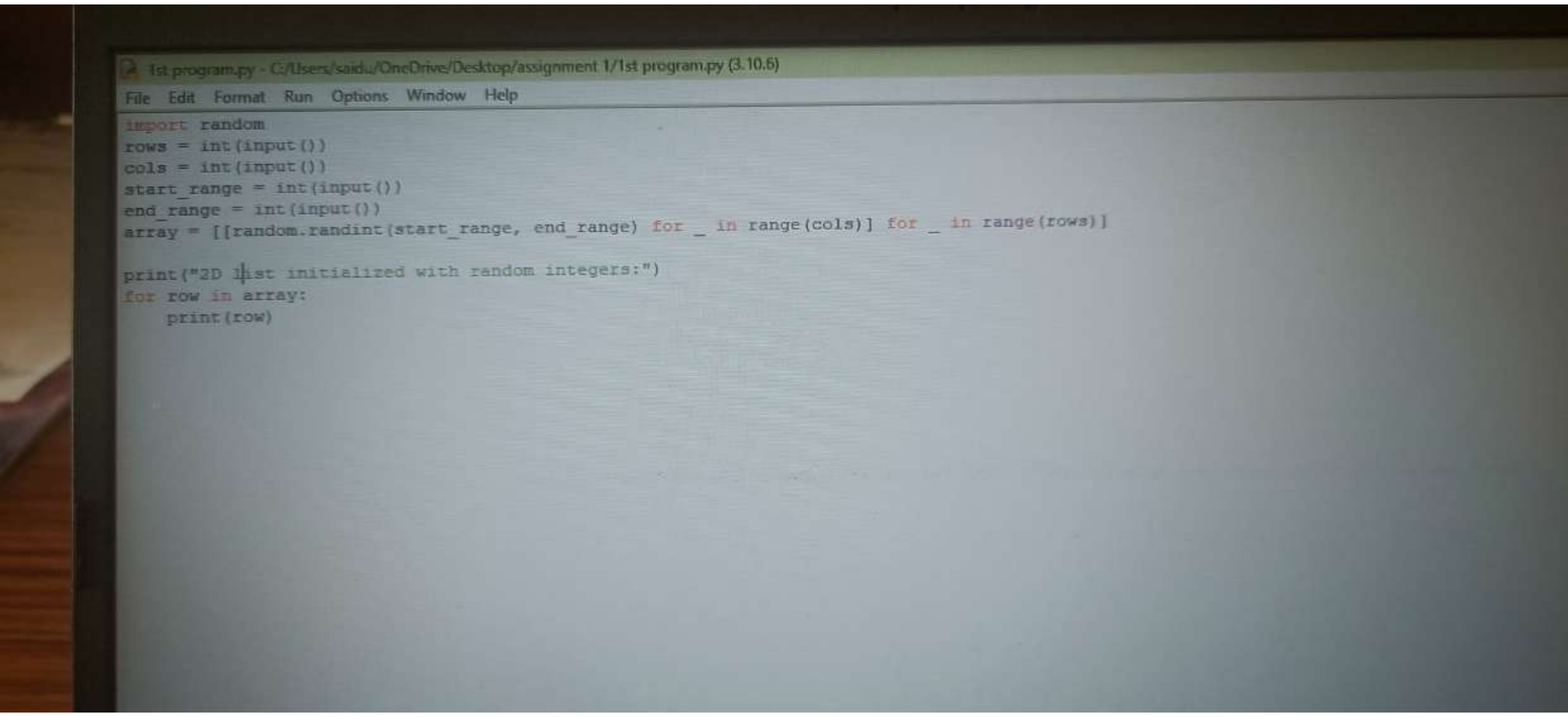
```
    [5, 5, 5]
    [5, 5, 5]
    [5, 5, 5]
>>>
    ================================================ RESTART: C:/Users/saidu/On
    7
    6
    4
    Array of evenly spaced numbers over the specified range:
    [7.0, 6.666666666666667, 6.333333333333333, 6.0]
>>> |
```


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

```
start = float(input())
end = float(input())
num_samples = int(input())
step = (end - start) / (num_samples - 1)
array = [start + step * i for i in range(num_samples)]
print("Array of evenly spaced numbers over the specified range:")
print(array)
```



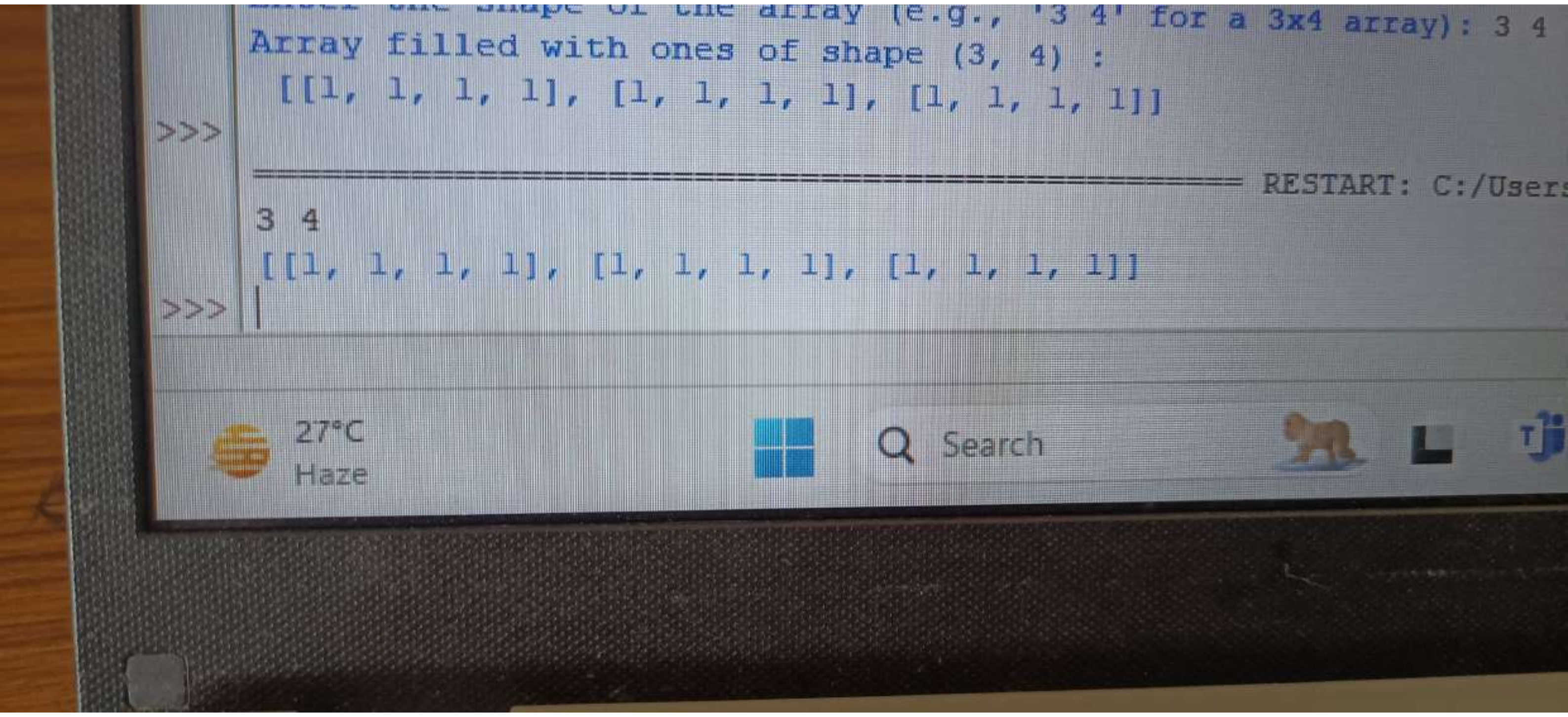
```
[5, 5, 5]
>>> ============================================= RESTART: C:/Users/saidu/On
3
3
5
5
2D list initialized with random integers:
[5, 5, 5]
[5, 5, 5]
[5, 5, 5]
>>> 
```



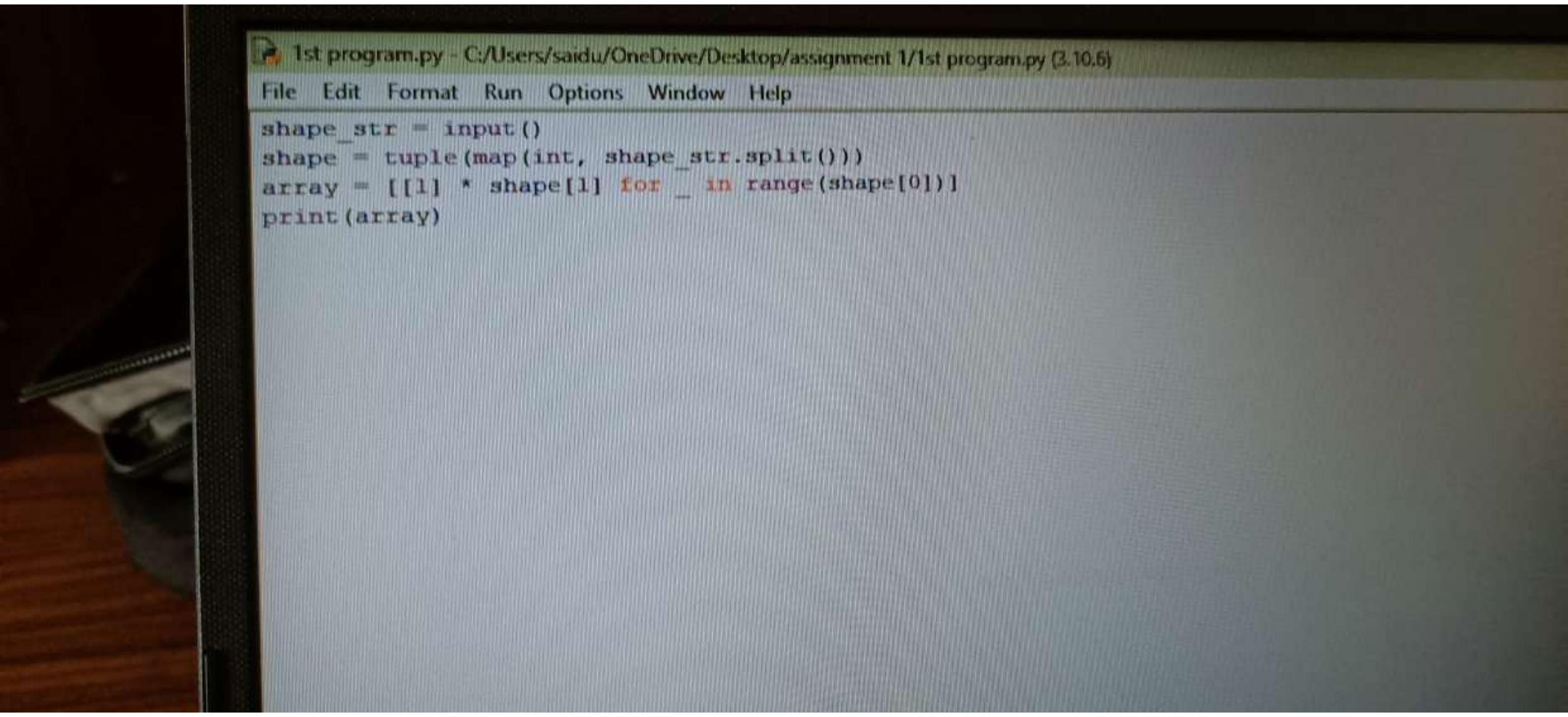
1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)

```
import random
rows = int(input())
cols = int(input())
start_range = int(input())
end_range = int(input())
array = [[random.randint(start_range, end_range) for _ in range(cols)] for _ in range(rows)]

print("2D list initialized with random integers:")
for row in array:
    print(row)
```

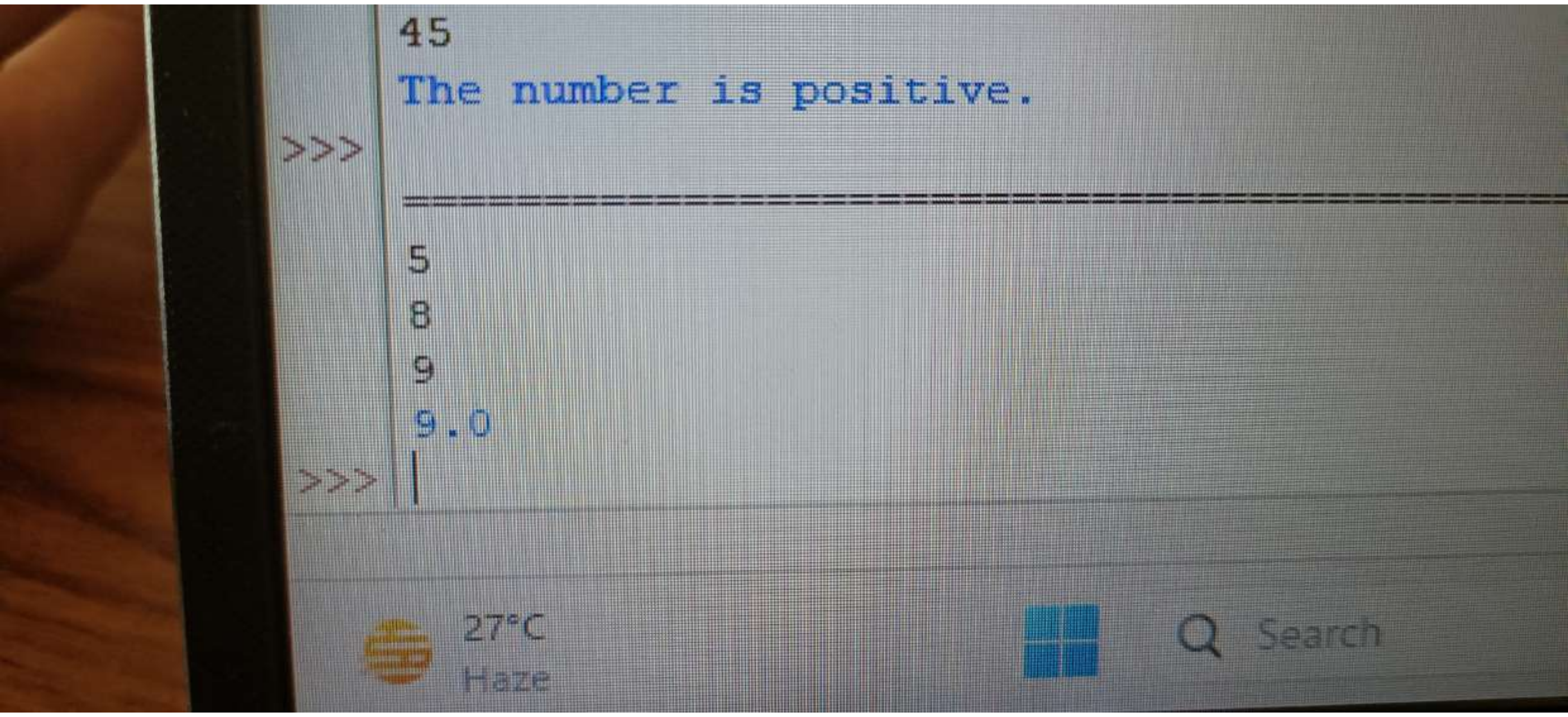


```
Enter the shape of the array (e.g., '3 4' for a 3x4 array): 3 4
Array filled with ones of shape (3, 4) :
 [[1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]]
>>>
=================================================== RESTART: C:/Users
3 4
[[1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]]
>>> |
```

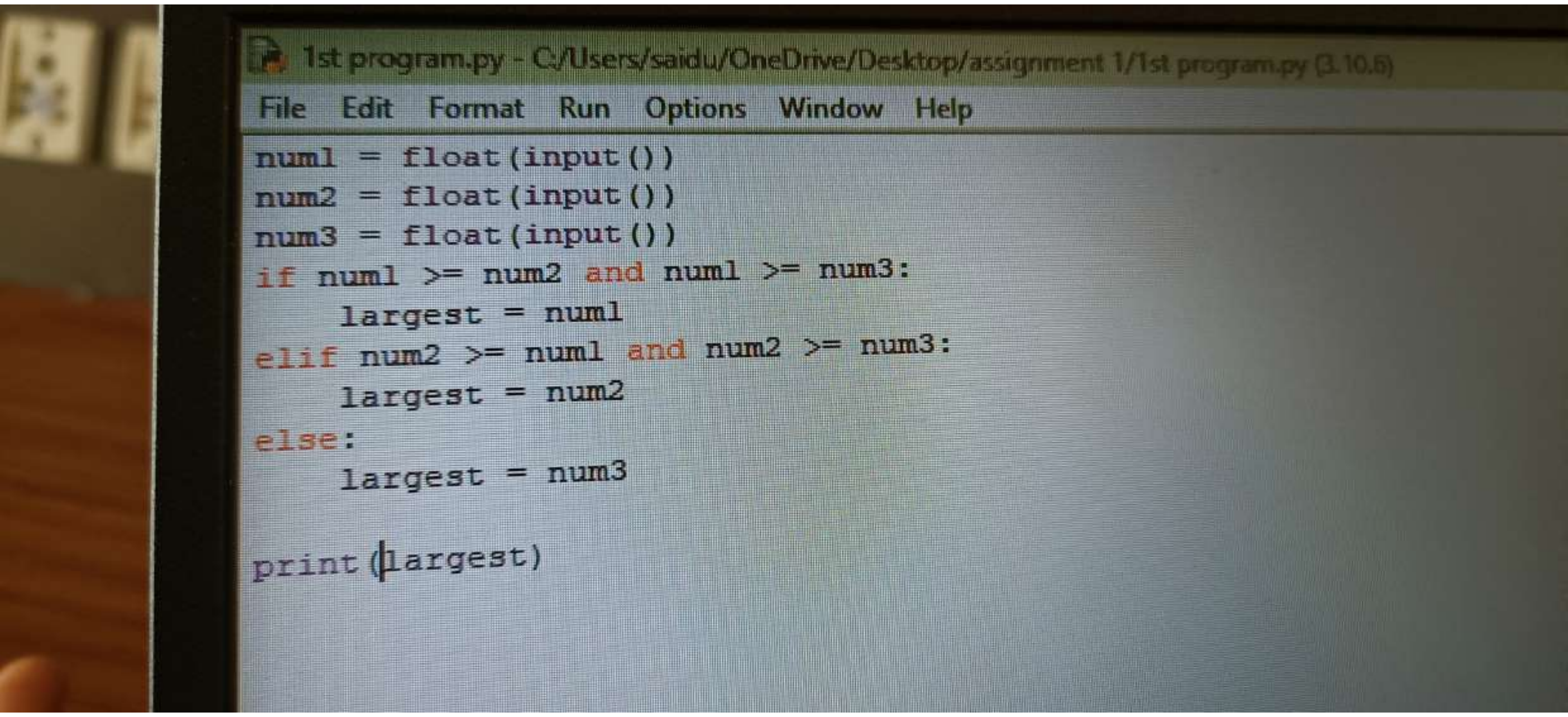


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

```
shape_str = input()
shape = tuple(map(int, shape_str.split()))
array = [[1] * shape[1] for _ in range(shape[0])]
print(array)
```



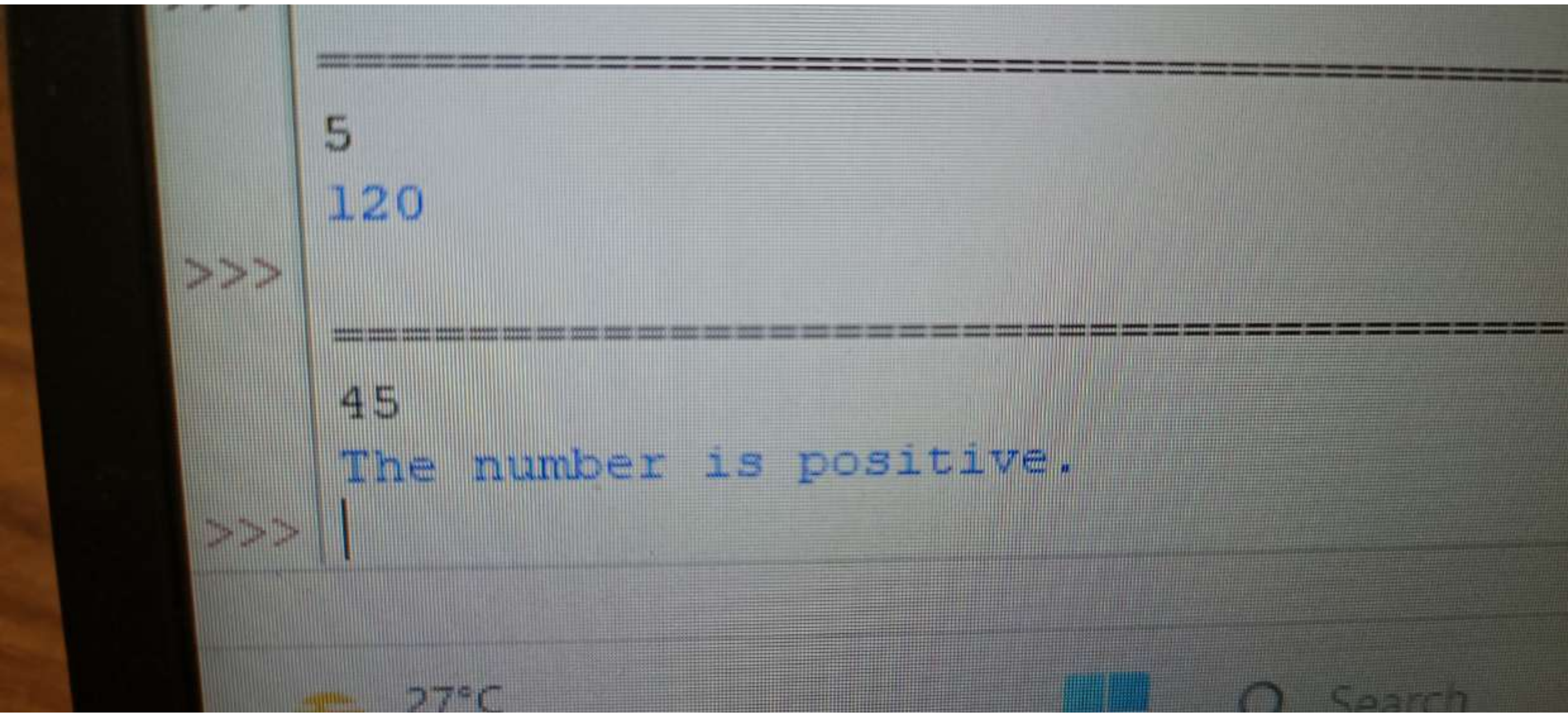
```
    45
    The number is positive.
>>>
    ===================================================
    5
    8
    9
    9.0
>>> 
```



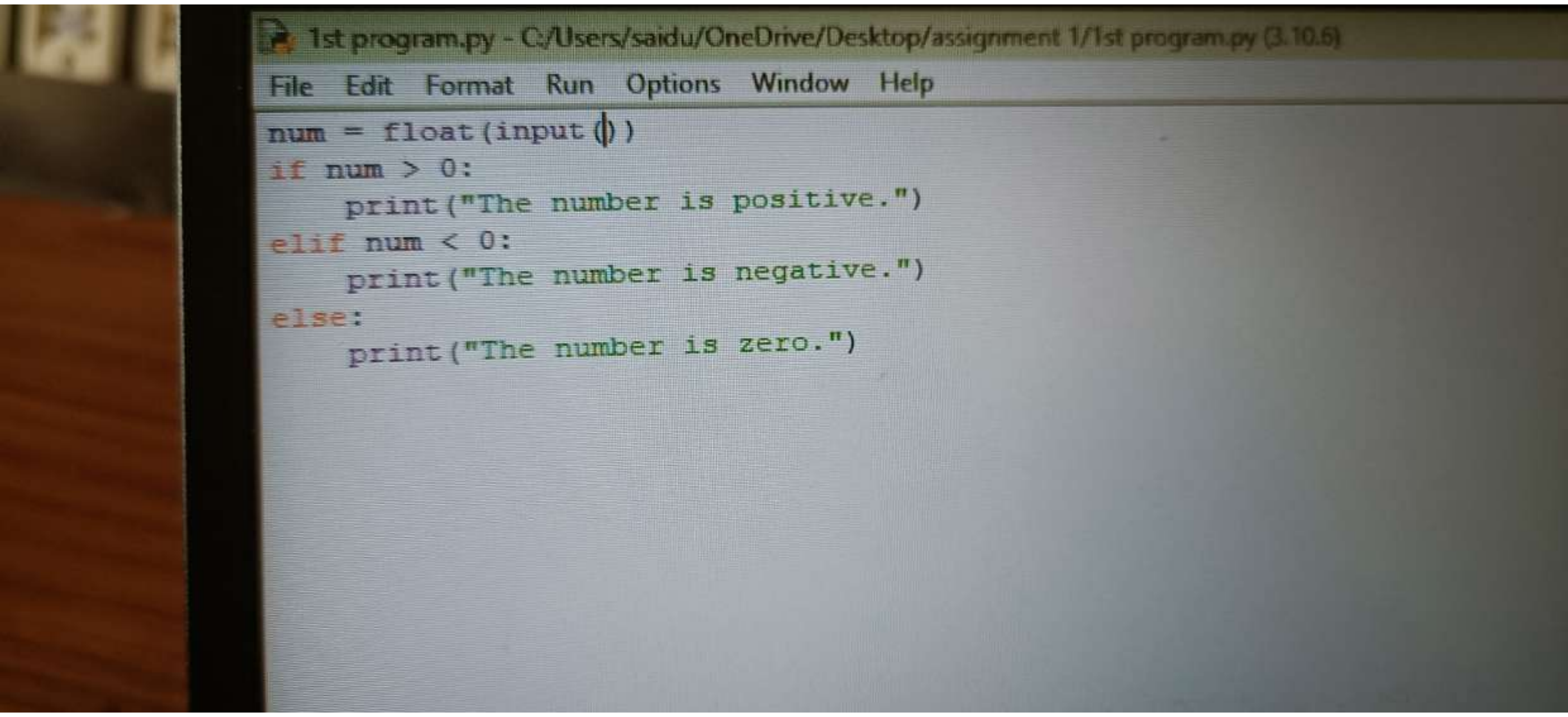
1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)

```
num1 = float(input())
num2 = float(input())
num3 = float(input())
if num1 >= num2 and num1 >= num3:
    largest = num1
elif num2 >= num1 and num2 >= num3:
    largest = num2
else:
    largest = num3

print(largest)
```

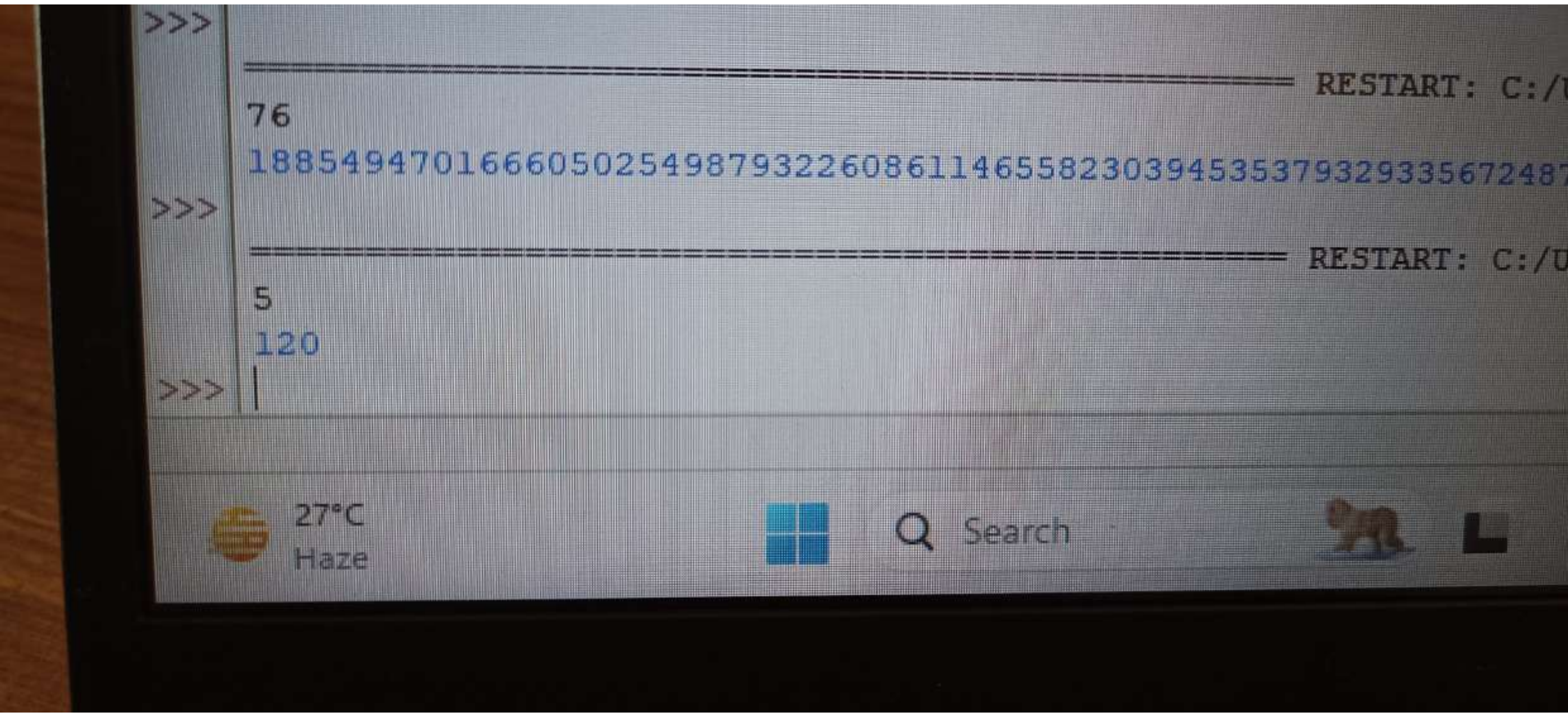


```
==================================================================
5
120
>>>
==================================================================
45
The number is positive.
>>> |
```

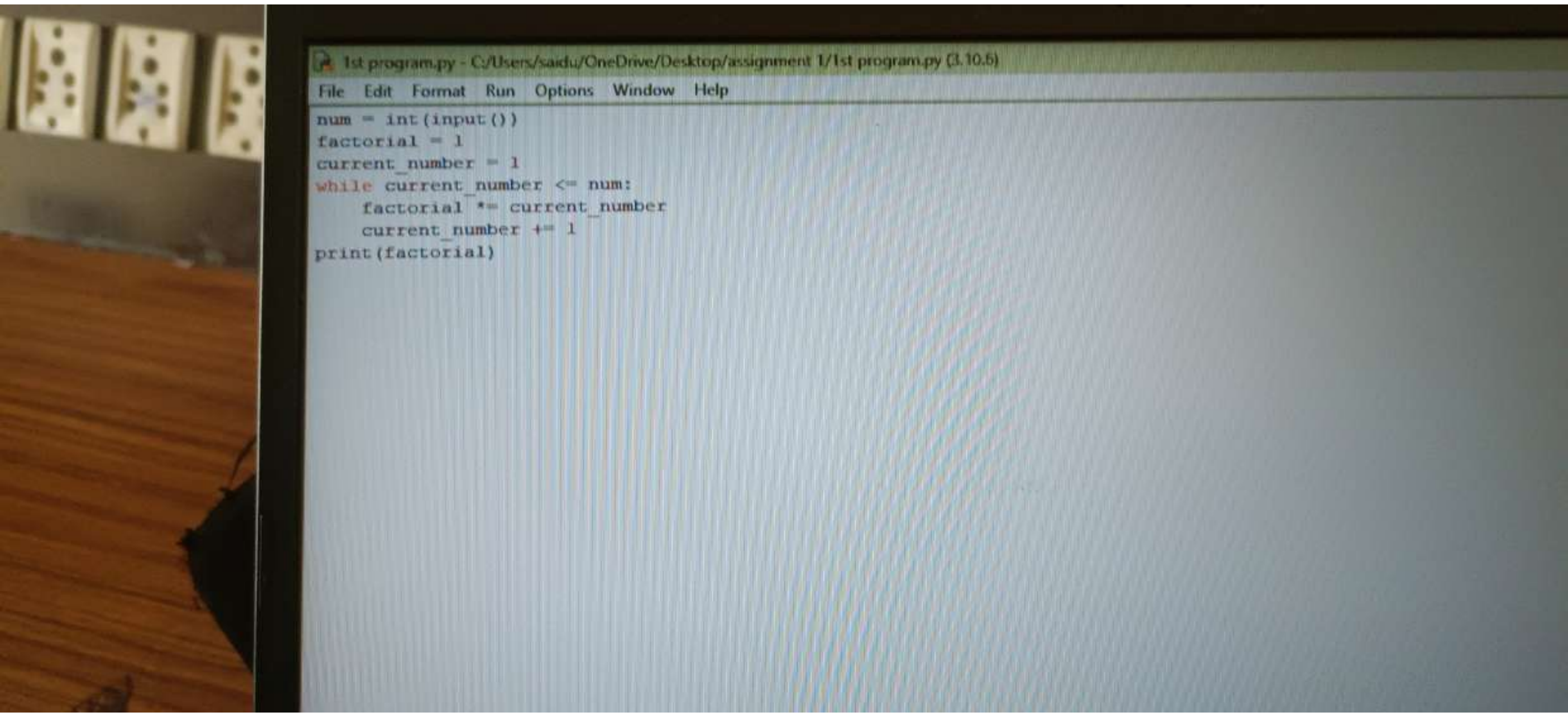


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File
Edit
Format
Run
Options
Window
Help

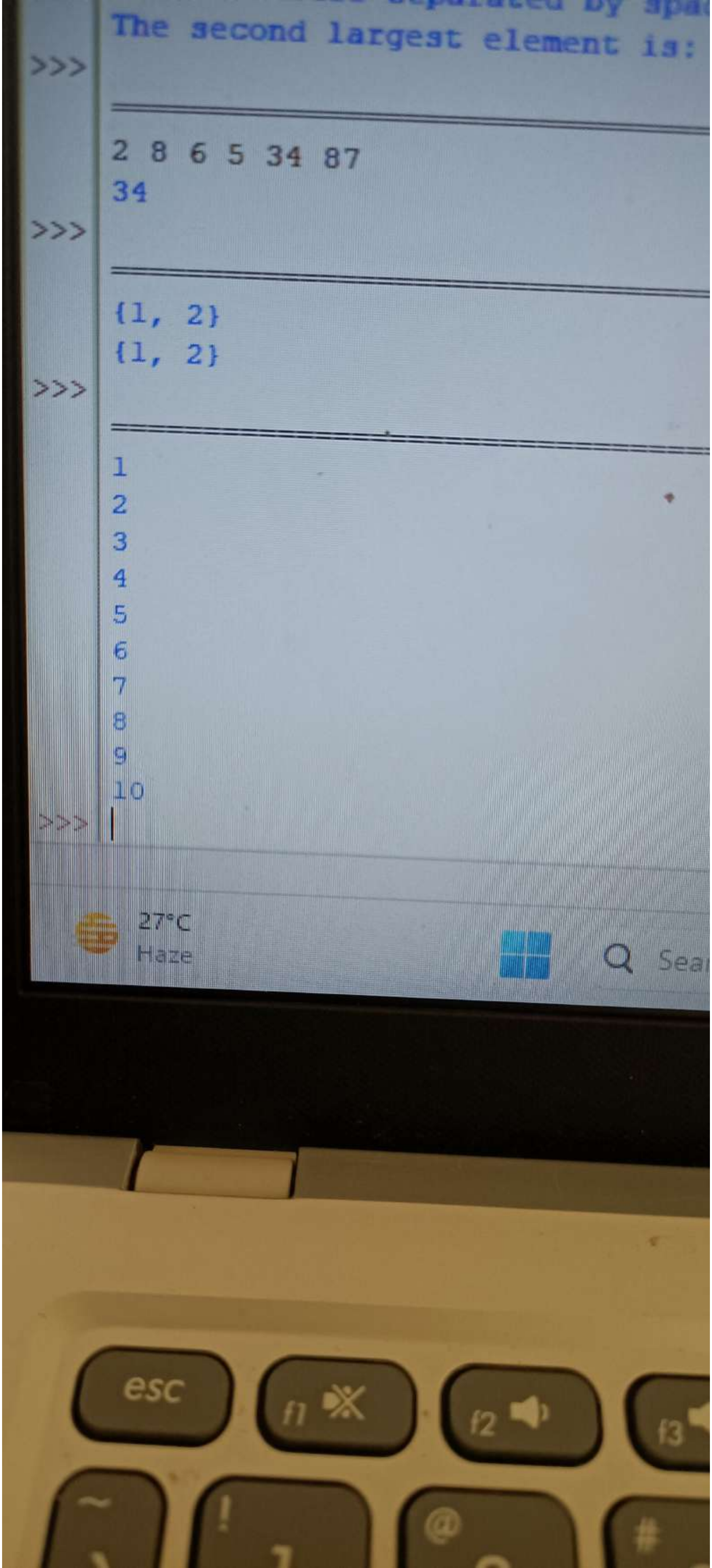
```
num = float(input())
if num > 0:
    print("The number is positive.")
elif num < 0:
    print("The number is negative.")
else:
    print("The number is zero.")
```



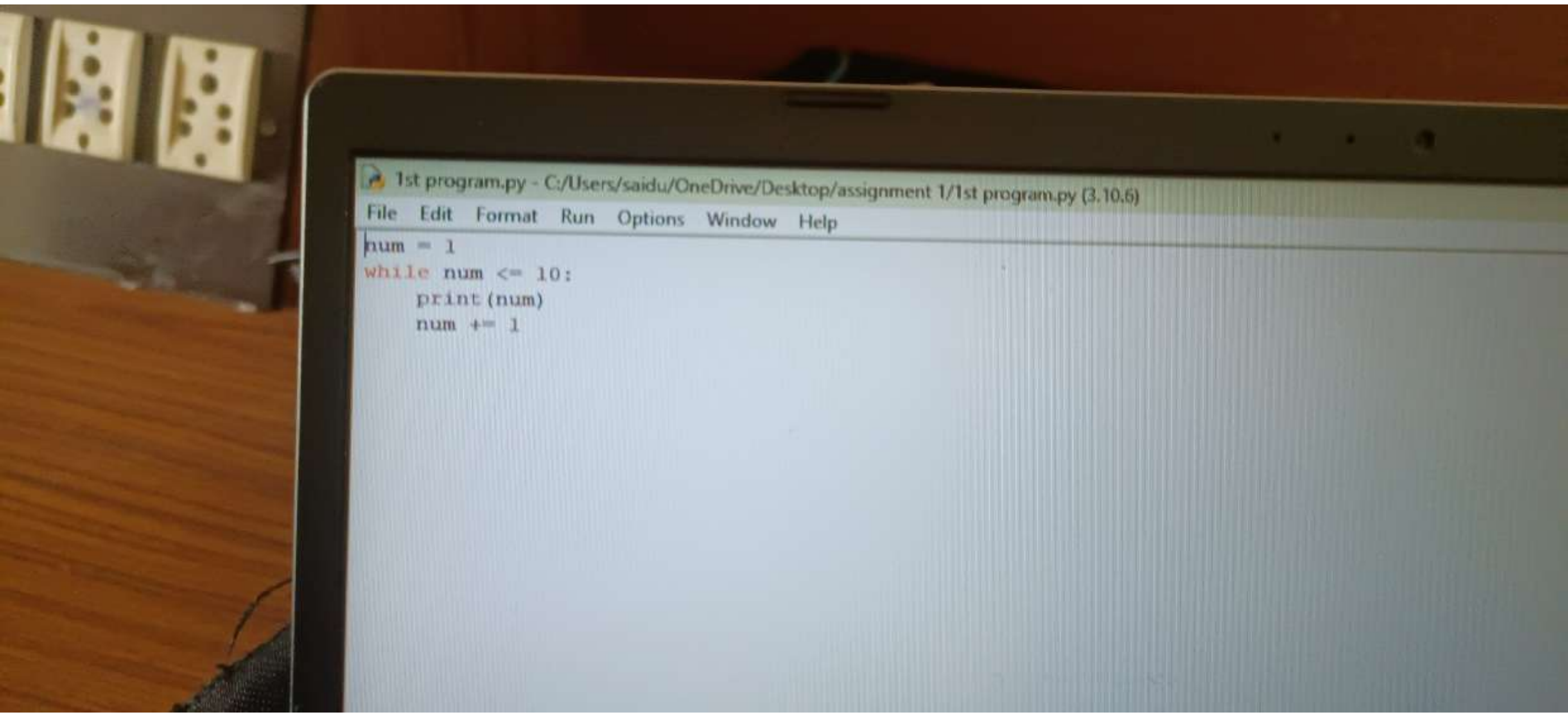
```
>>>
============================================================ RESTART: C:/U
76
1885494701666050254987932260861146558230394535379329335672487
>>>
============================================================ RESTART: C:/U
5
120
>>> |
```


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

```
num = int(input())
factorial = 1
current_number = 1
while current_number <= num:
    factorial *= current_number
    current_number += 1
print(factorial)
```

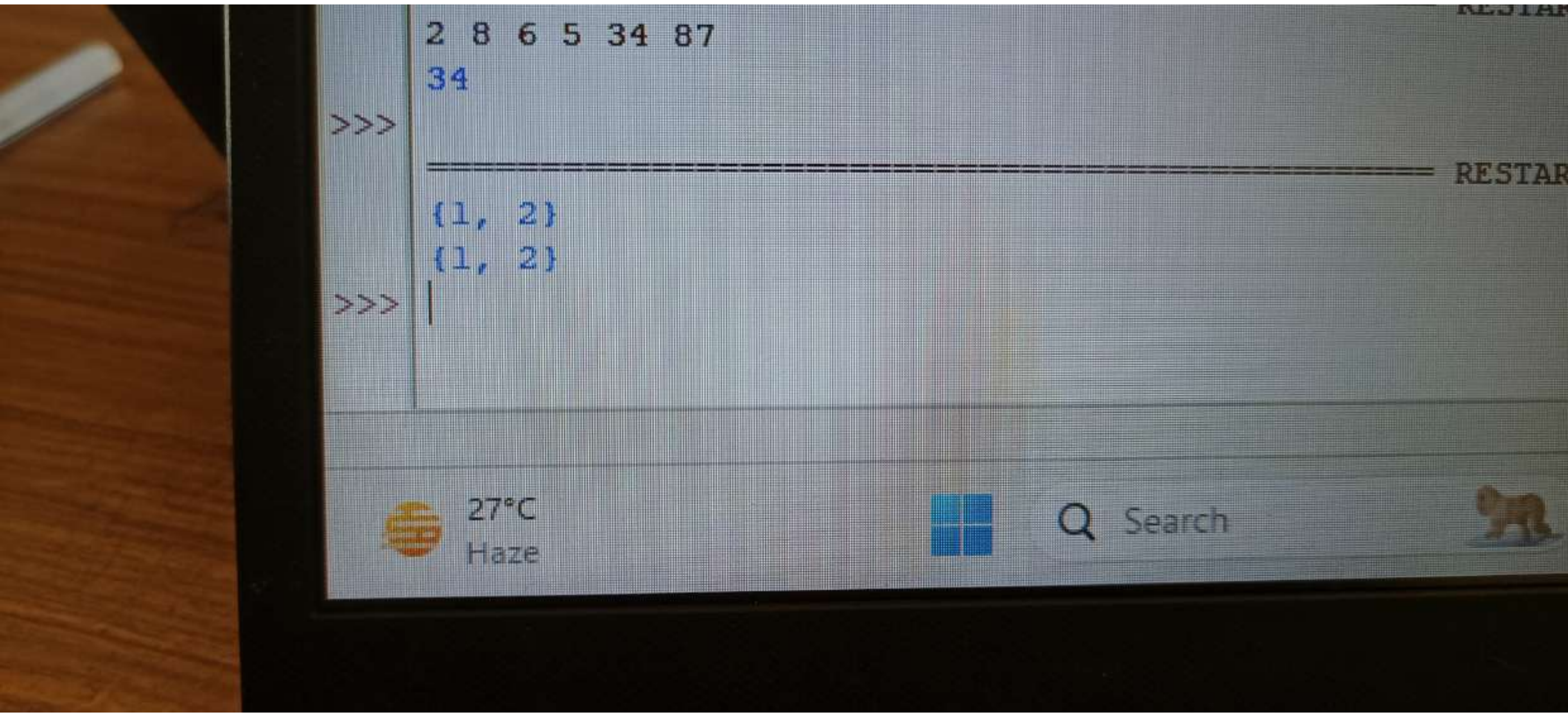


```
                    separated by spa
    The second largest element is:
>>>
    ================================
    2 8 6 5 34 87
    34
>>>
    ================================
    {1, 2}
    {1, 2}
>>>
    ================================
    1
    2
    3
    4
    5
    6
    7
    8
    9
    10
>>> |
```

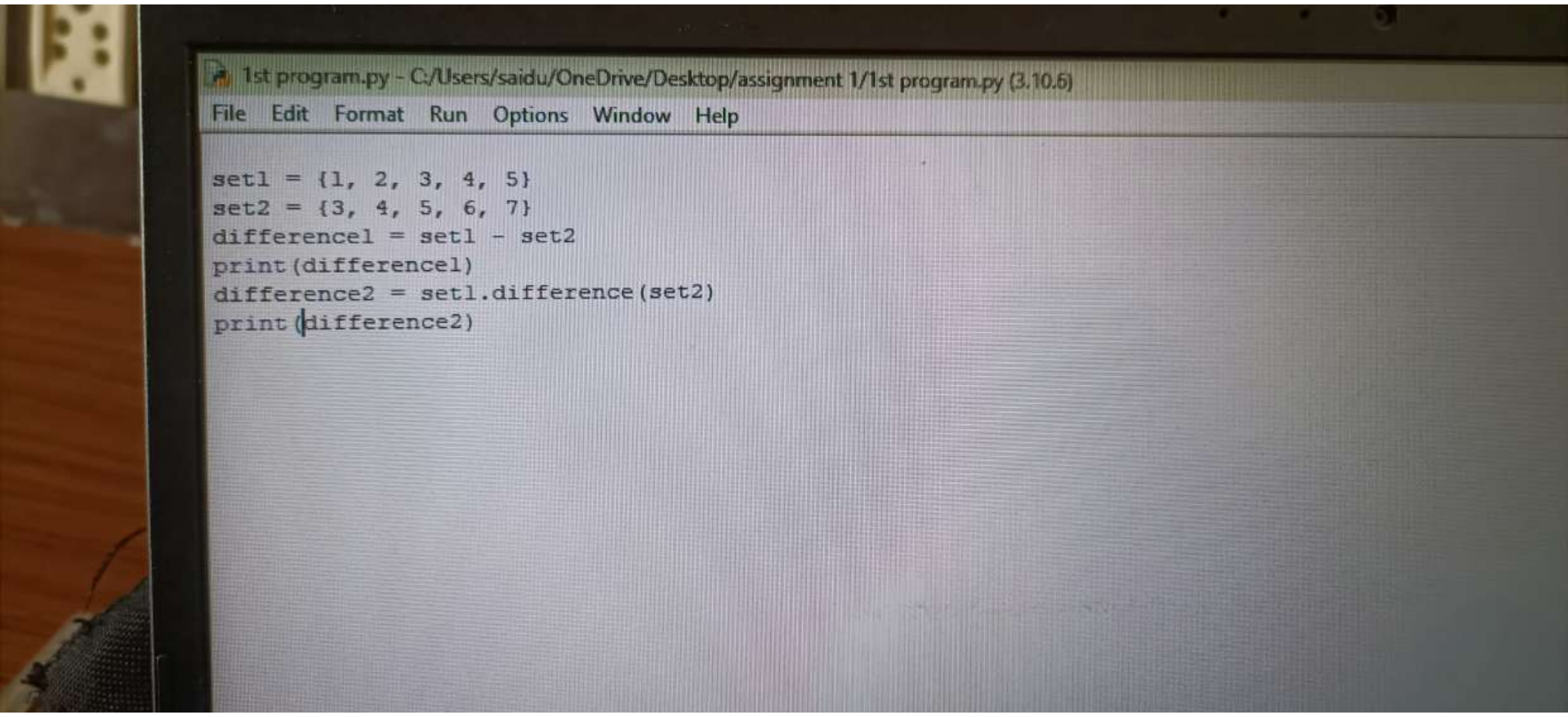


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

```
num = 1
while num <= 10:
    print(num)
    num += 1
```

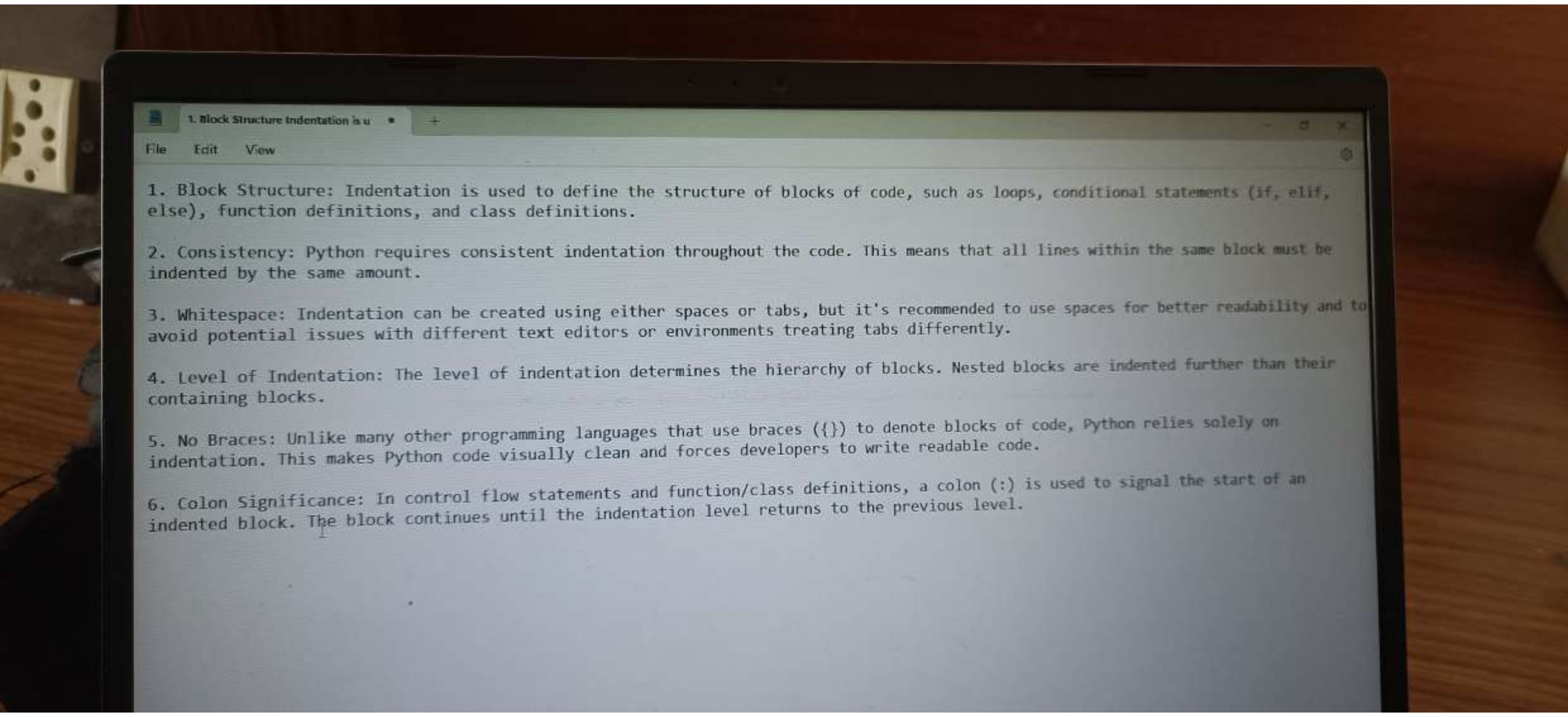


```
2 8 6 5 34 87
34
>>>
======================================= RESTAR
{1, 2}
{1, 2}
>>>
```



1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

```
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 5, 6, 7}
difference1 = set1 - set2
print(difference1)
difference2 = set1.difference(set2)
print(difference2)
```



1. Block Structure: Indentation is used to define the structure of blocks of code, such as loops, conditional statements (if, elif, else), function definitions, and class definitions.

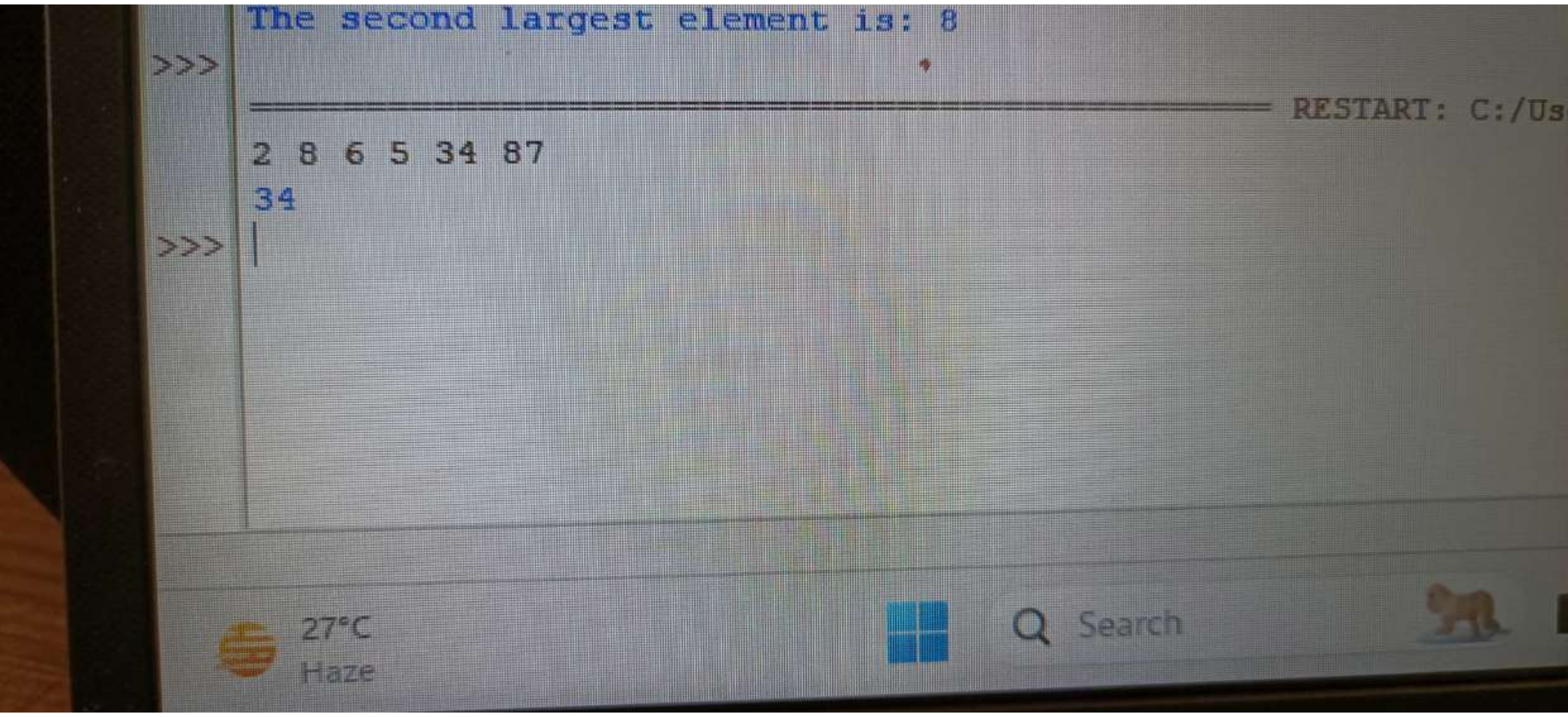
2. Consistency: Python requires consistent indentation throughout the code. This means that all lines within the same block must be indented by the same amount.

3. Whitespace: Indentation can be created using either spaces or tabs, but it's recommended to use spaces for better readability and to avoid potential issues with different text editors or environments treating tabs differently.

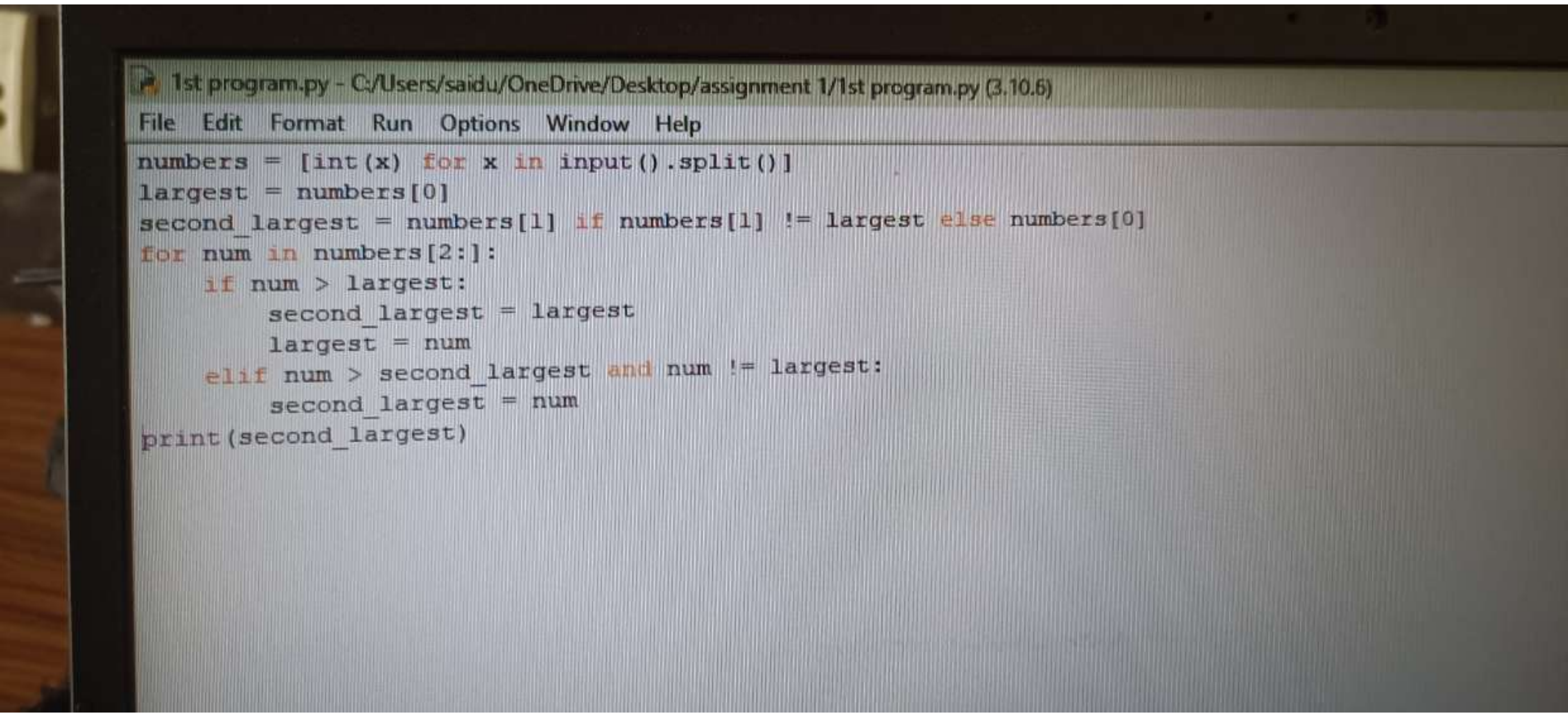
4. Level of Indentation: The level of indentation determines the hierarchy of blocks. Nested blocks are indented further than their containing blocks.

5. No Braces: Unlike many other programming languages that use braces ({}) to denote blocks of code, Python relies solely on indentation. This makes Python code visually clean and forces developers to write readable code.

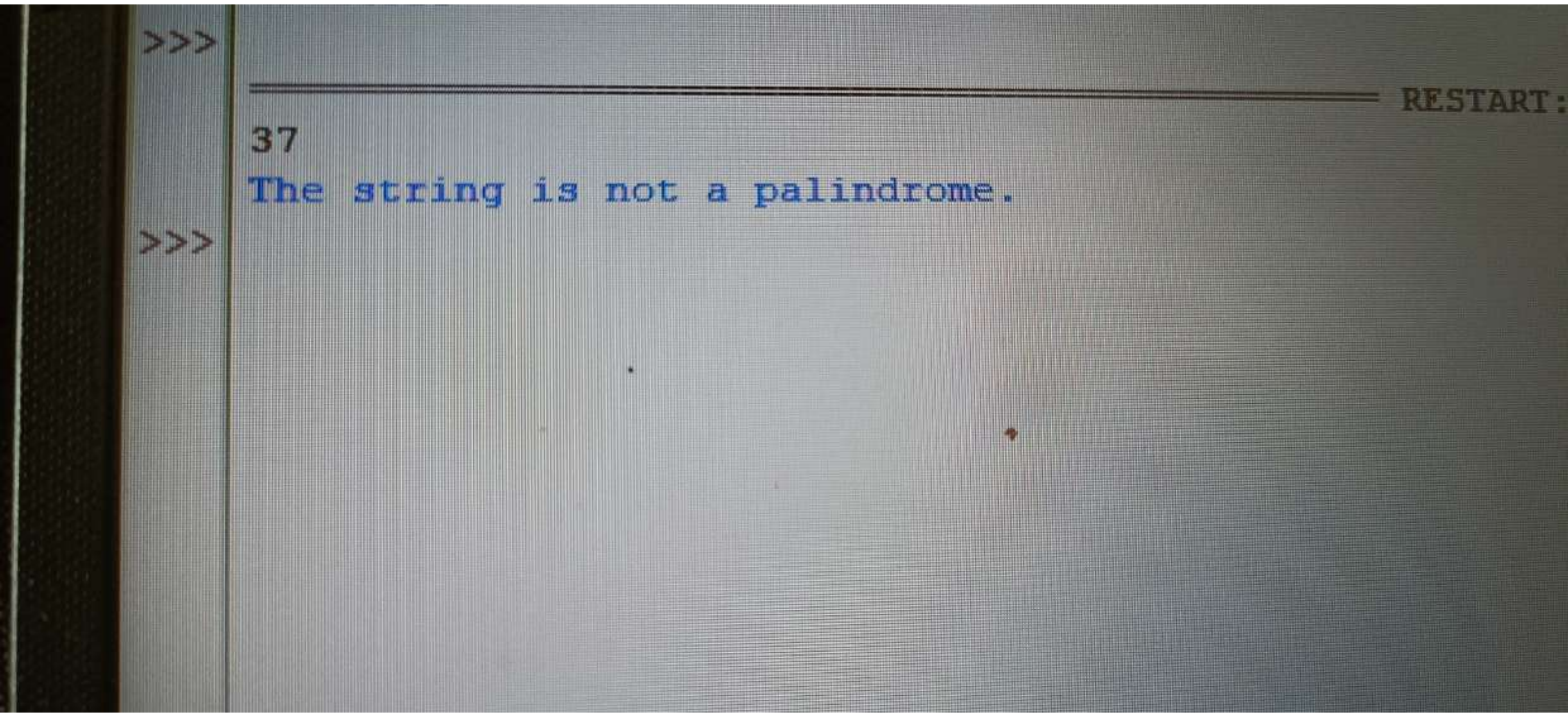
6. Colon Significance: In control flow statements and function/class definitions, a colon (:) is used to signal the start of an indented block. The block continues until the indentation level returns to the previous level.



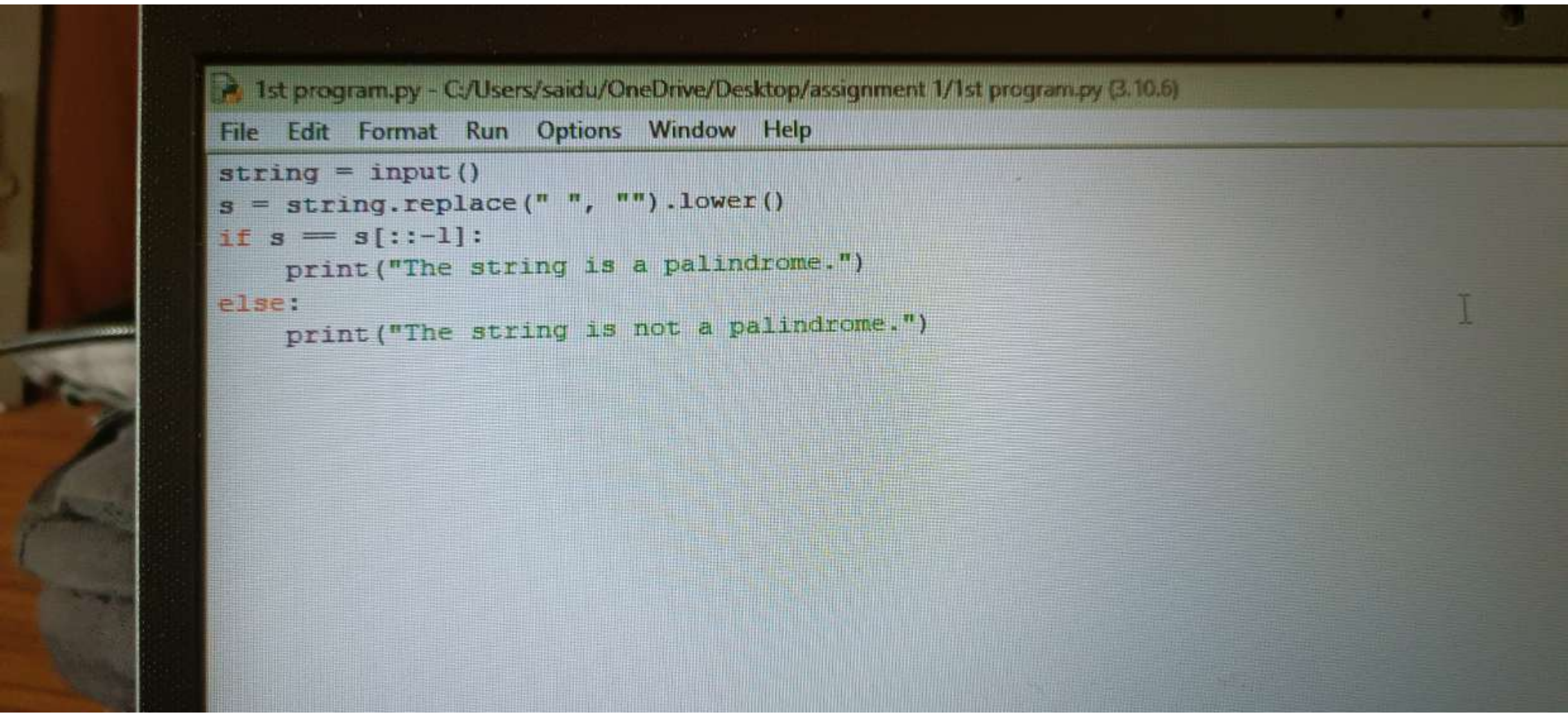
```
The second largest element is: 8
>>>
=============================================== RESTART: C:/Us
2 8 6 5 34 87
34
>>>
```


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

```
numbers = [int(x) for x in input().split()]
largest = numbers[0]
second_largest = numbers[1] if numbers[1] != largest else numbers[0]
for num in numbers[2:]:
    if num > largest:
        second_largest = largest
        largest = num
    elif num > second_largest and num != largest:
        second_largest = num
print(second_largest)
```

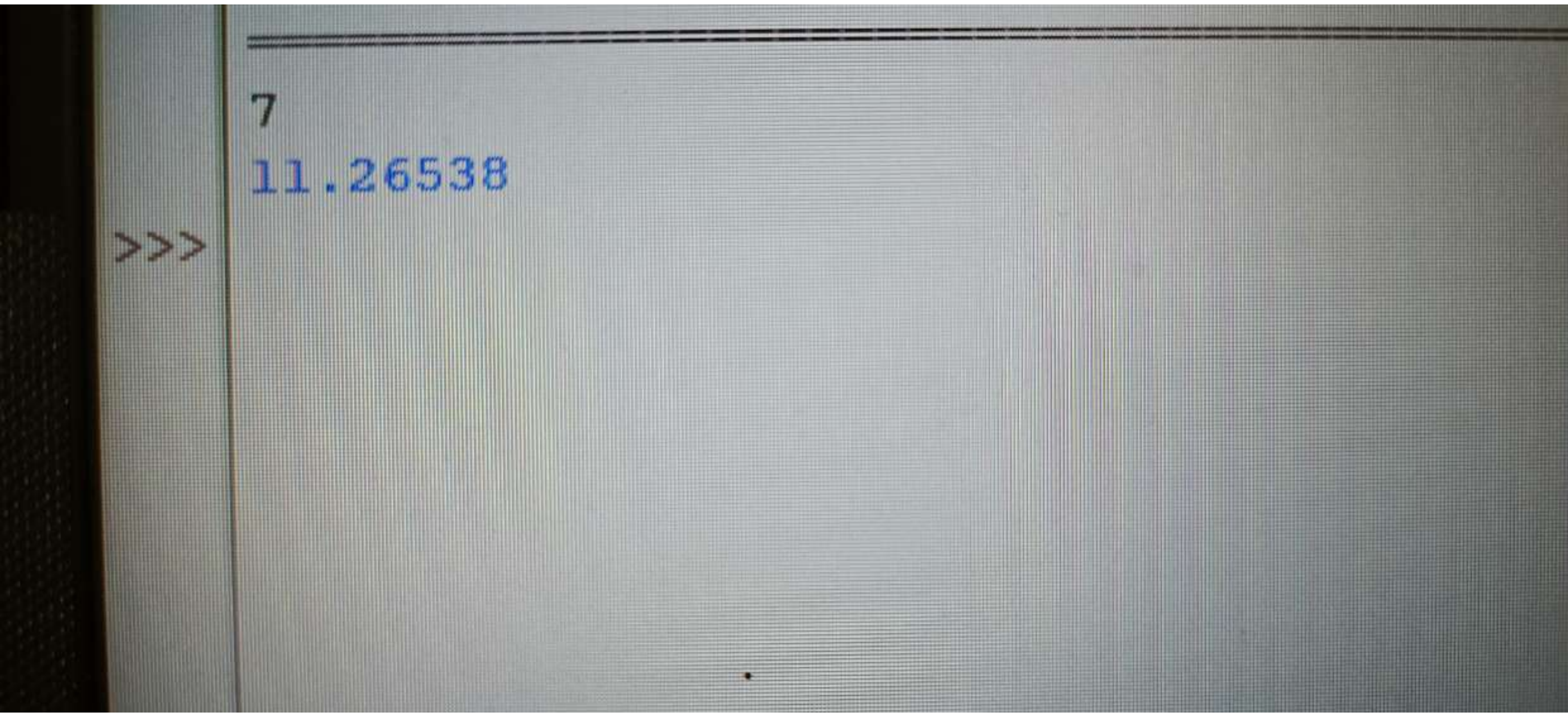


```
>>>
==================================================================== RESTART:
37
The string is not a palindrome.
>>>
```

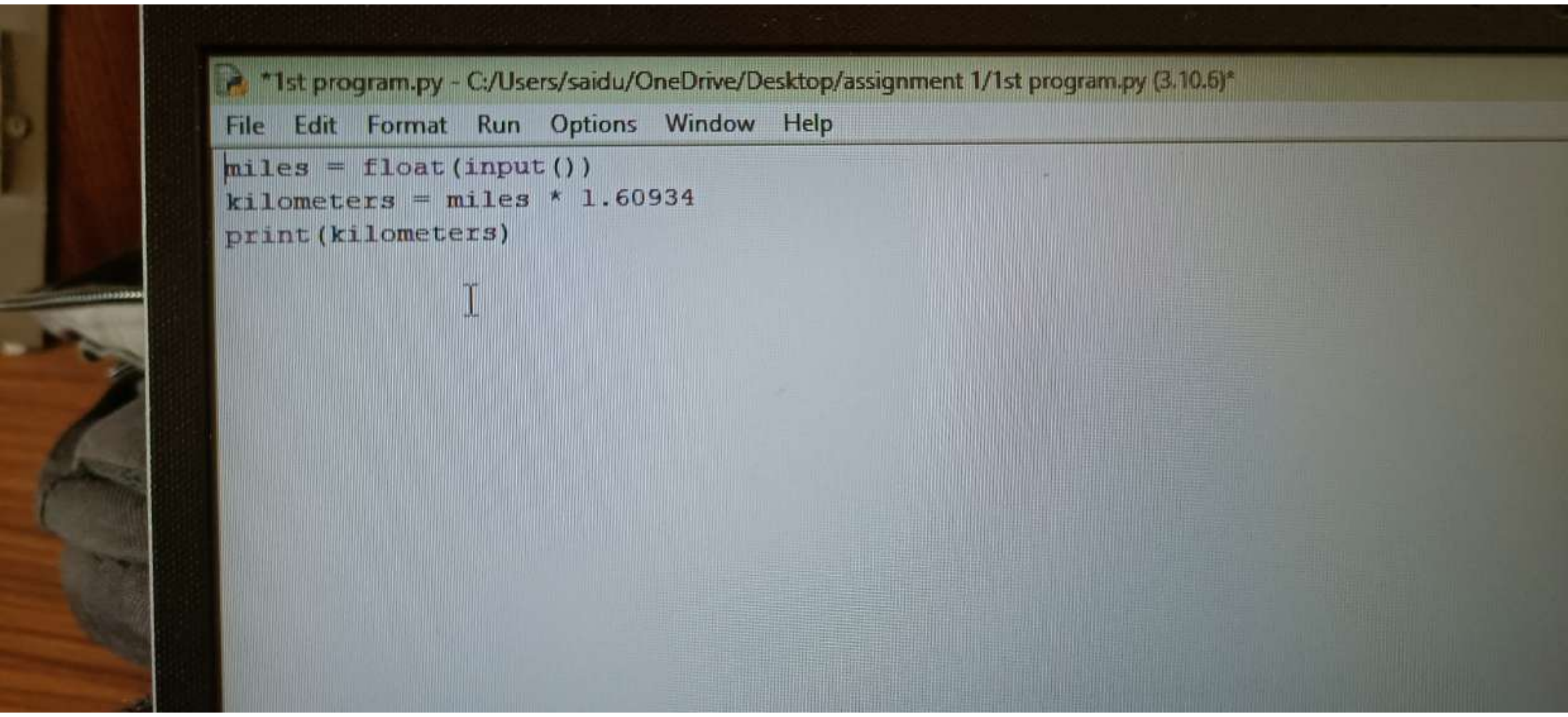


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

```
string = input()
s = string.replace(" ", "").lower()
if s == s[::-1]:
    print("The string is a palindrome.")
else:
    print("The string is not a palindrome.")
```

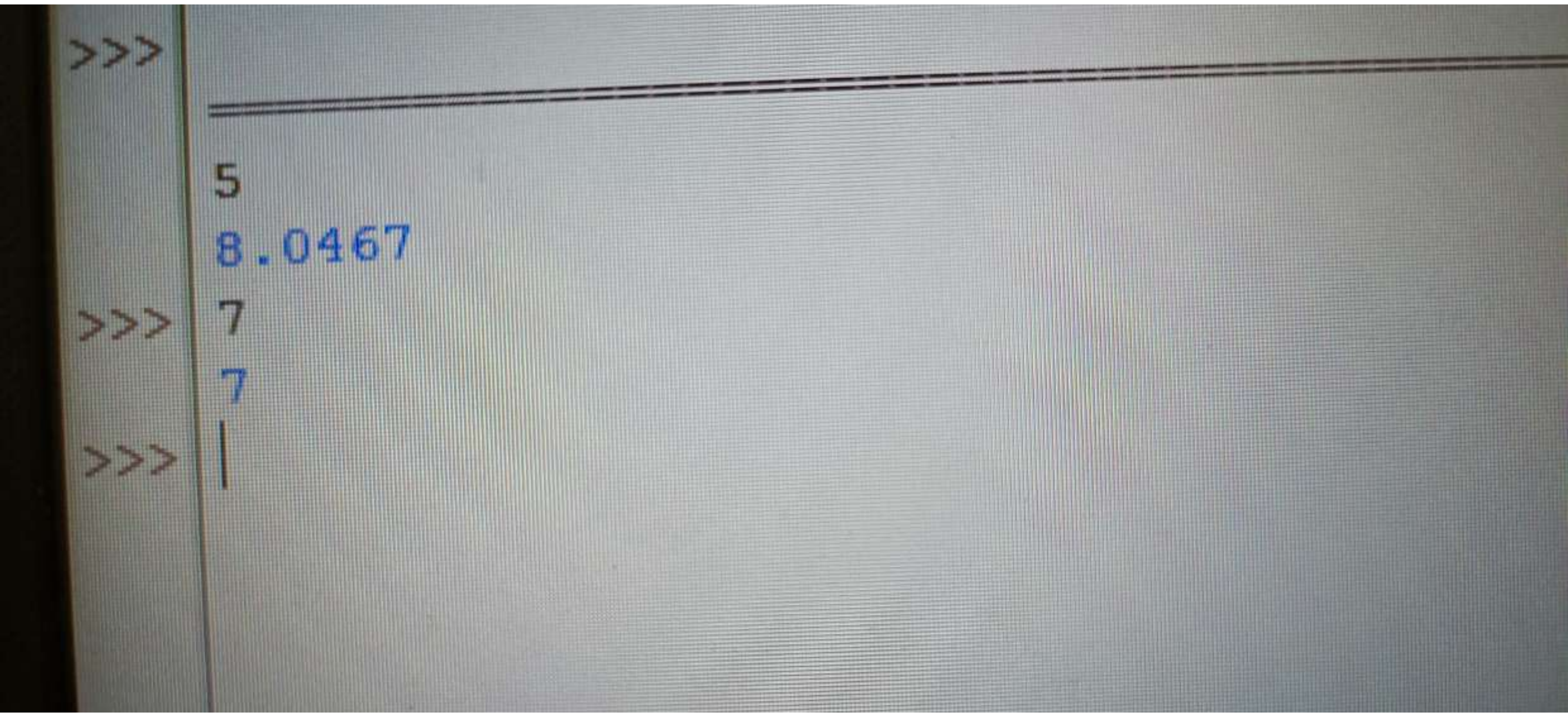


```
7
11.26538
>>>
```

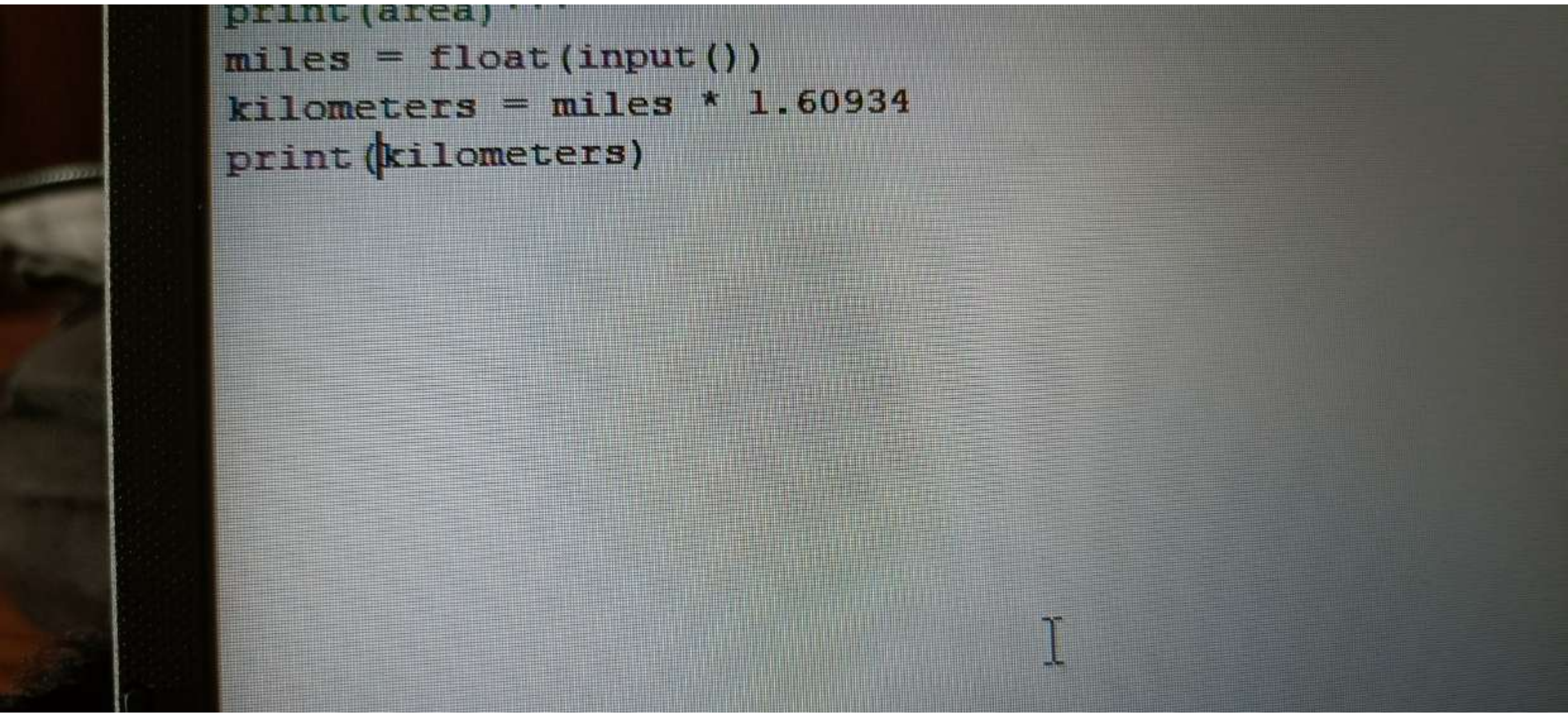


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File Edit Format Run Options Window Help

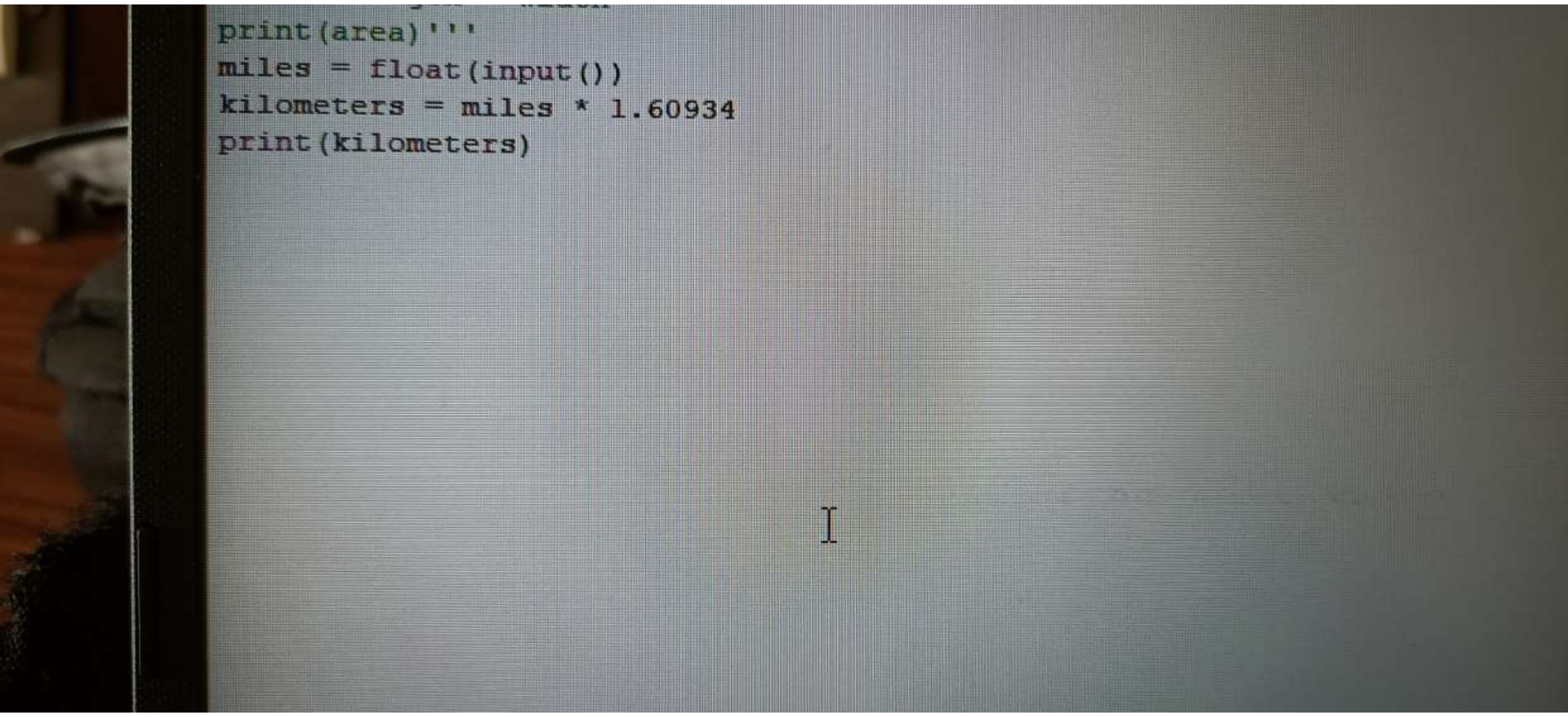
```
miles = float(input())
kilometers = miles * 1.60934
print(kilometers)
```



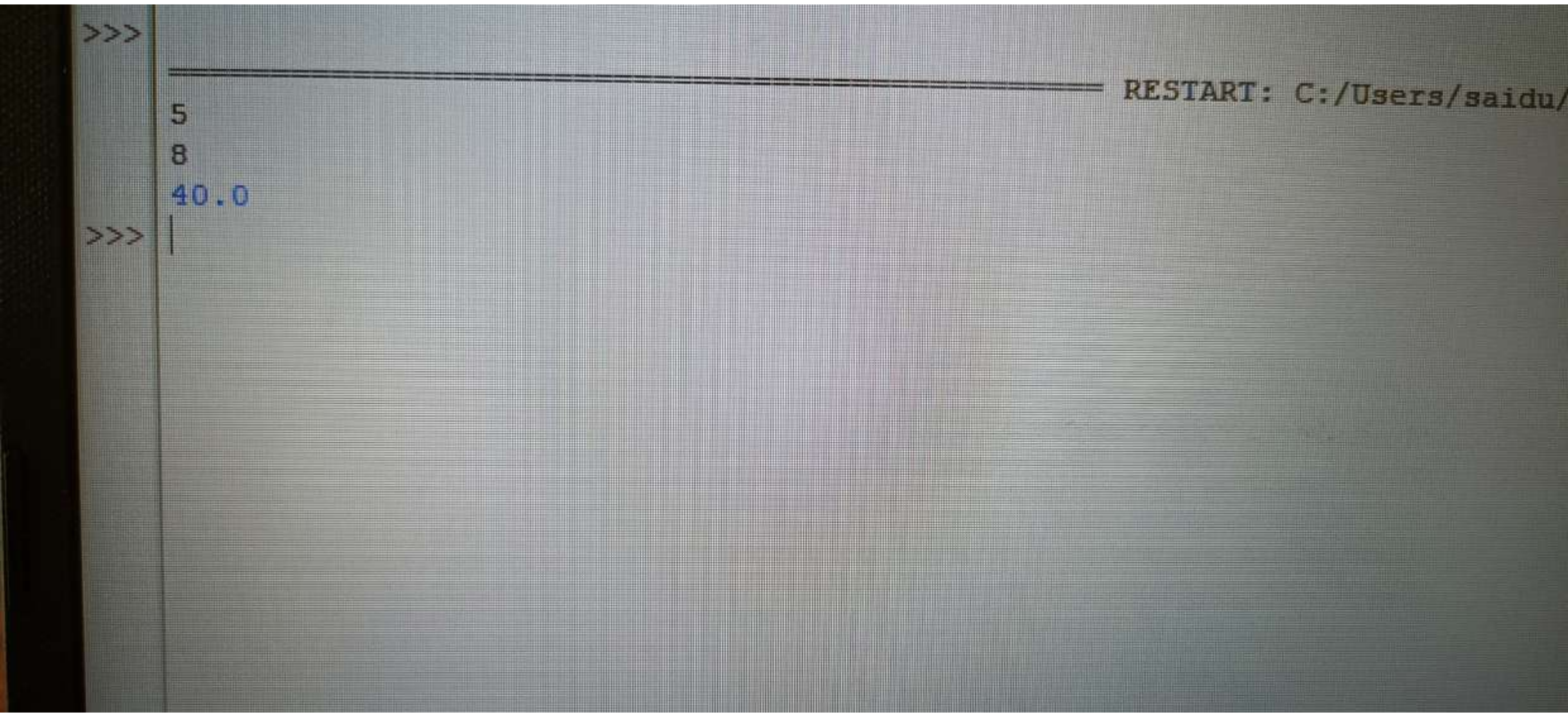
```
>>>
=============================
5
8.0467
>>> 7
7
>>> 
```



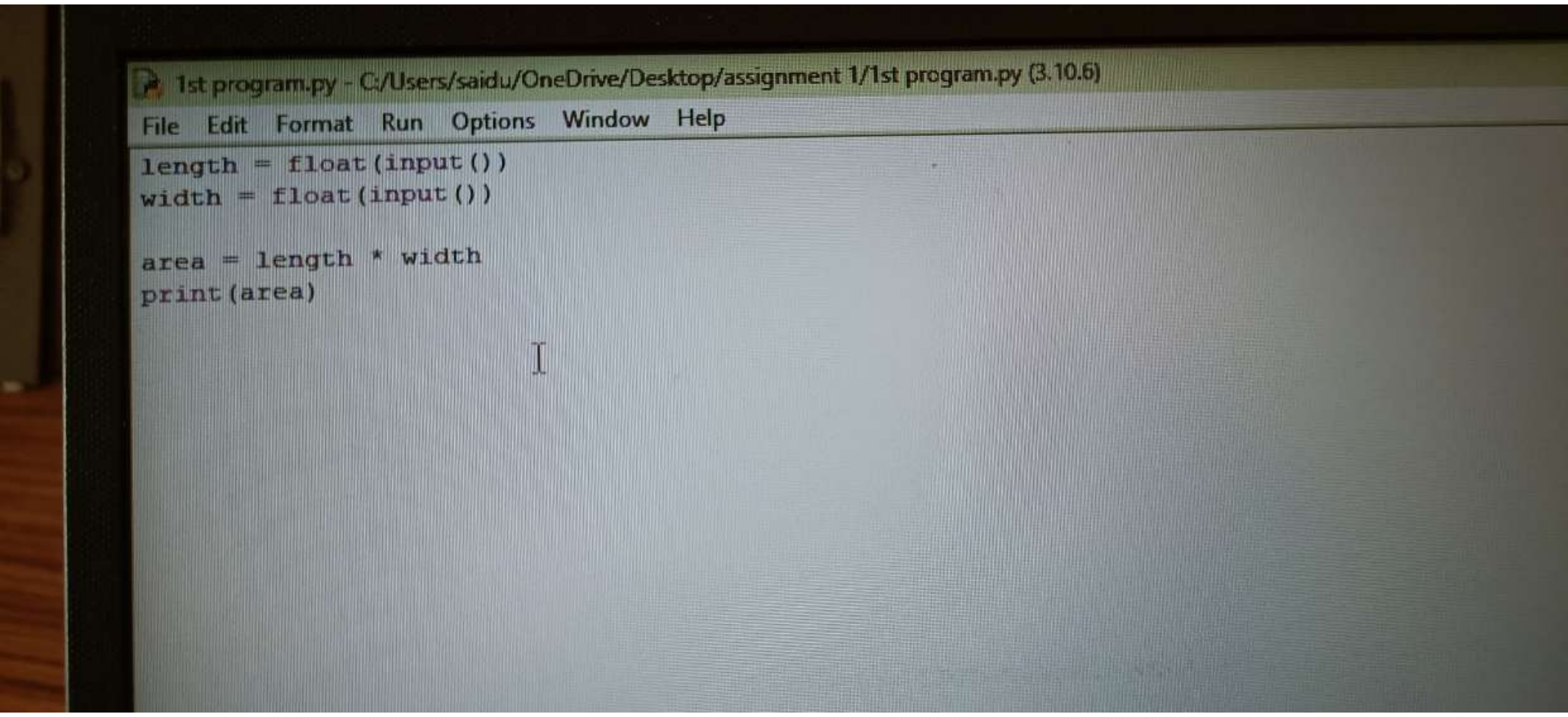
```
print(area)'''
miles = float(input())
kilometers = miles * 1.60934
print(kilometers)
```



```
print(area)'''
miles = float(input())
kilometers = miles * 1.60934
print(kilometers)
```



```
>>>
=============================================== RESTART: C:/Users/saidu/
5
8
40.0
>>> |
```


1st program.py - C:/Users/saidu/OneDrive/Desktop/assignment 1/1st program.py (3.10.6)
File
Edit
Format
Run
Options
Window
Help

```
length = float(input())
width = float(input())

area = length * width
print(area)
```