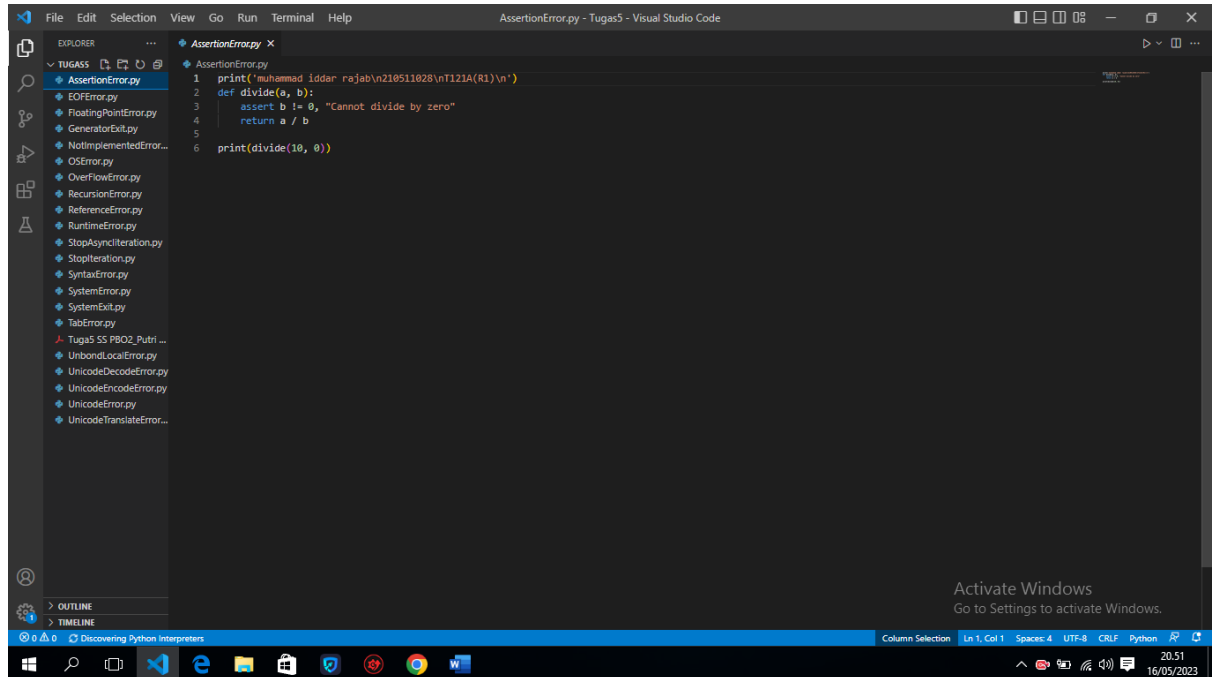


Nama : muhammad iddar rajab

Kelas : R1

Nim : 210511028

1. assertionError

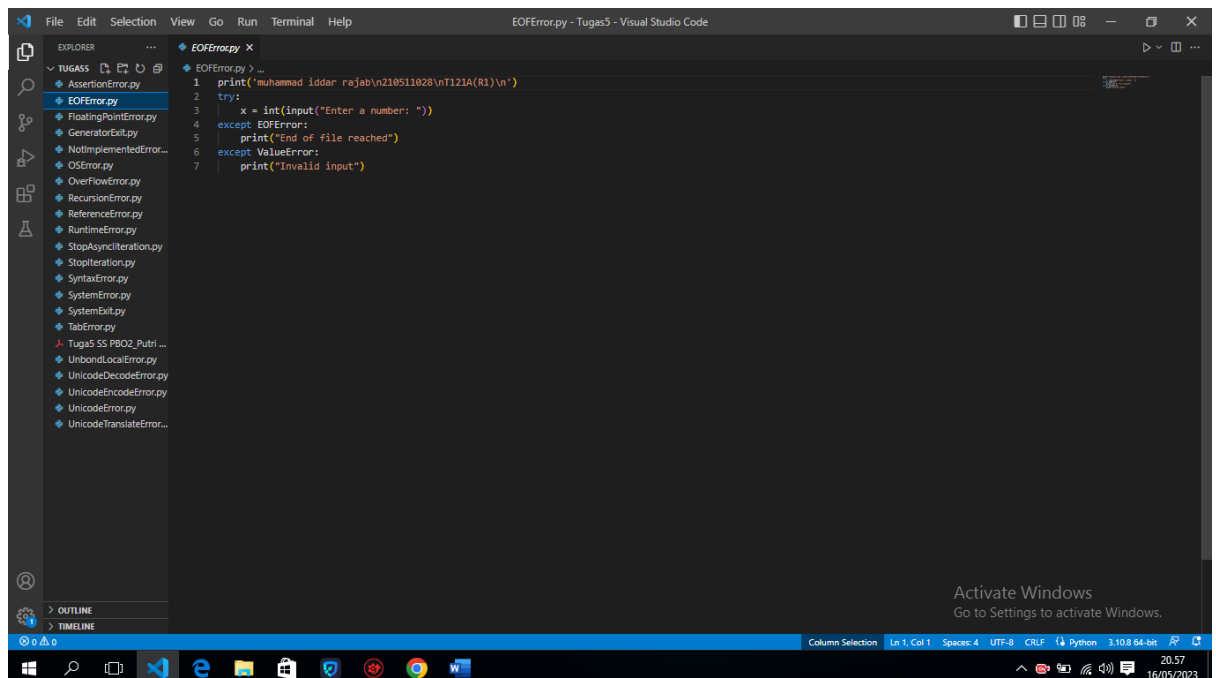


The screenshot shows the Visual Studio Code interface with a file named `AssertionError.py` open. The code in the editor is as follows:

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 def divide(a, b):
3     assert b != 0, "Cannot divide by zero"
4     return a / b
5
6 print(divide(10, 0))
```

The Explorer sidebar on the left shows a project named `TUGAS3` with a folder `AssertionError.py` containing several Python error-related files. The status bar at the bottom indicates the file is using Python 3.10.8 64-bit.

2. EOFError

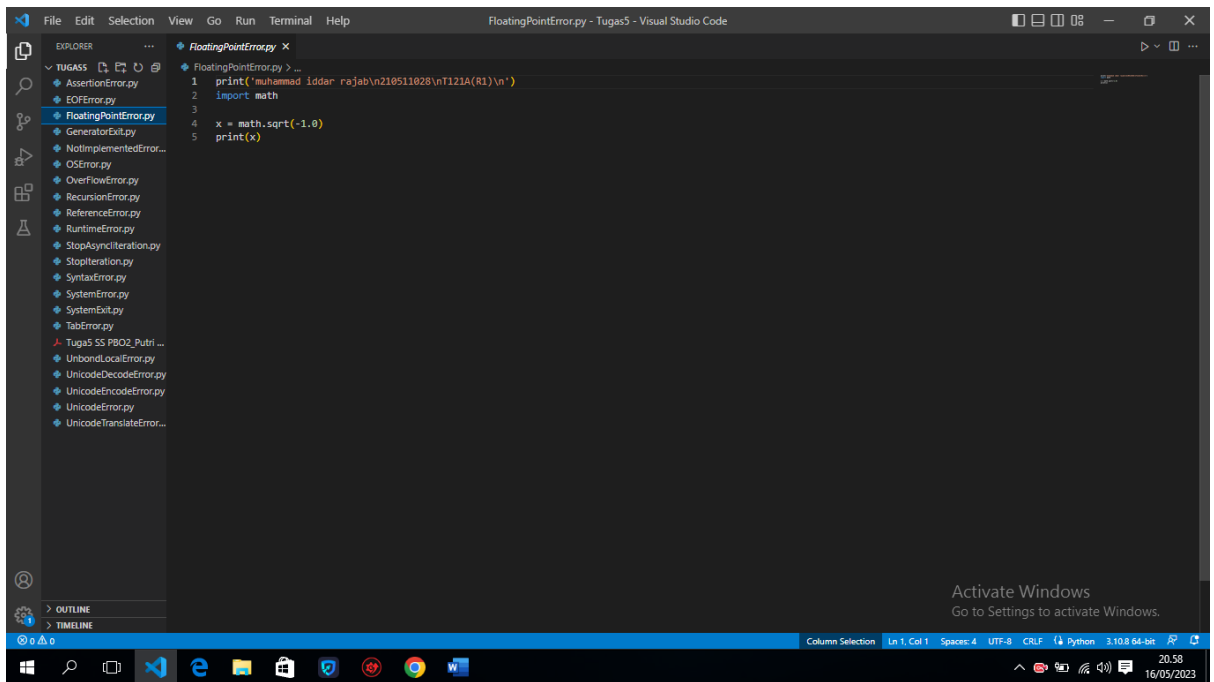


The screenshot shows the Visual Studio Code interface with a file named `EOFError.py` open. The code in the editor is as follows:

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 try:
3     x = int(input("Enter a number: "))
4 except EOFError:
5     print("End of file reached")
6 except ValueError:
7     print("Invalid input")
```

The Explorer sidebar on the left shows the same project structure as the first screenshot. The status bar at the bottom indicates the file is using Python 3.10.8 64-bit.

3. floatingPointError

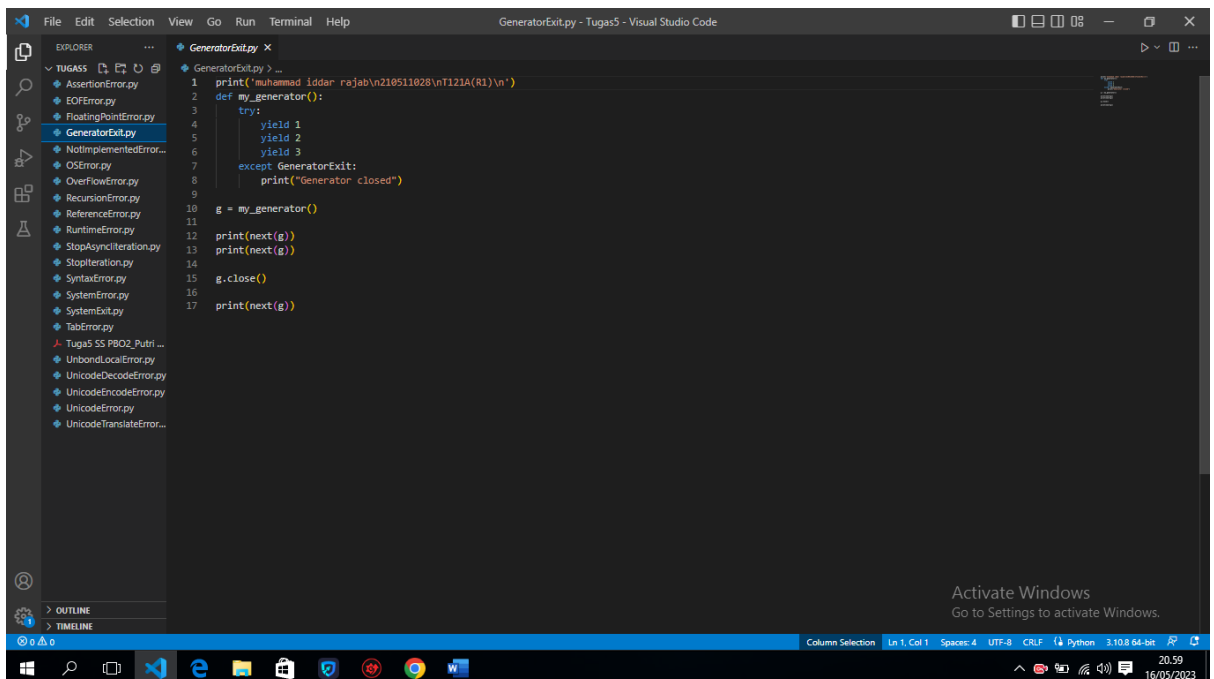


The screenshot shows the Visual Studio Code editor with the file `FloatingPointError.py` open. The Explorer sidebar on the left lists various Python error types under the `TUGAS5` folder, including `AssertionError.py`, `EOFError.py`, `FloatingPointError.py` (selected), `GeneratorExit.py`, `NotImplementedError.py`, `OSError.py`, `OverflowError.py`, `RecursionError.py`, `ReferenceError.py`, `RuntimeError.py`, `StopAsyncIteration.py`, `StopIteration.py`, `SyntaxError.py`, `SystemError.py`, `SystemExit.py`, and `TabError.py`. Below these are files for `Tugas5 SS PBO2_Putri ...` and several `Unicode` error types. The main editor window displays the code for `FloatingPointError.py`:

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 import math
3
4 x = math.sqrt(-1.0)
5 print(x)
```

The status bar at the bottom indicates the current position is `Ln 1, Col 1` with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the `Python` interpreter. The system clock shows `20:58` on `16/05/2023`.

4. GeneratorExit

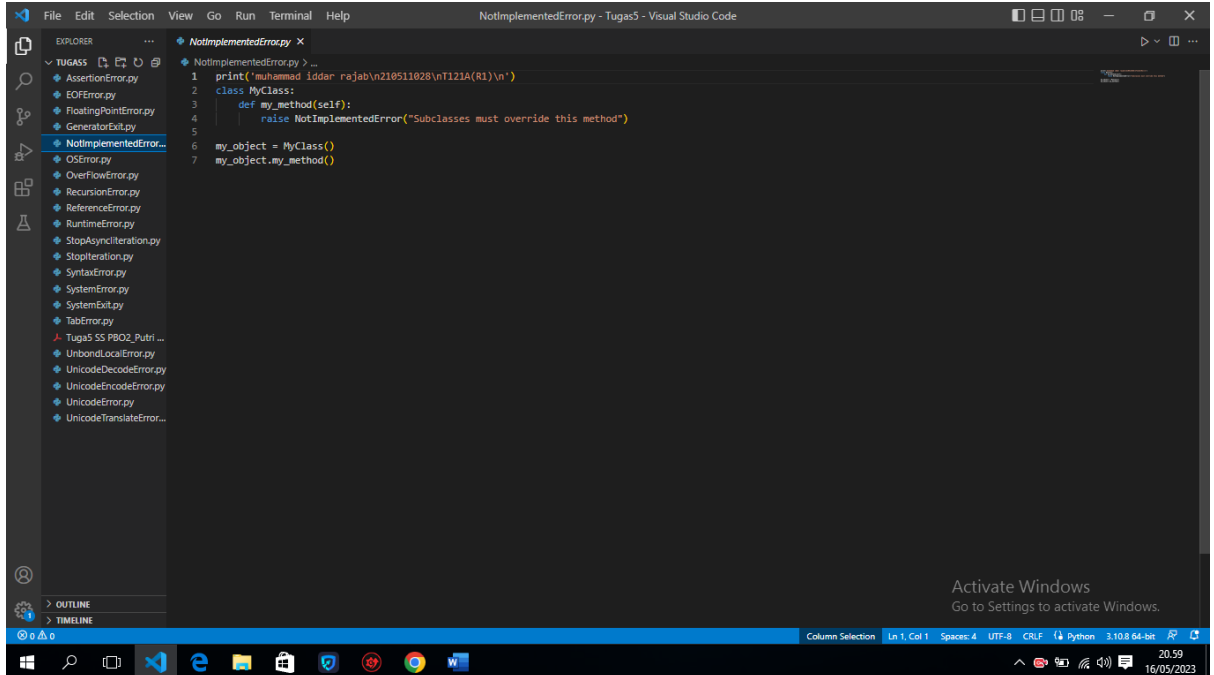


The screenshot shows the Visual Studio Code editor with the file `GeneratorExit.py` open. The Explorer sidebar on the left is identical to the previous screenshot, with `GeneratorExit.py` now selected. The main editor window displays the code for `GeneratorExit.py`:

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 def my_generator():
3     try:
4         yield 1
5         yield 2
6         yield 3
7     except GeneratorExit:
8         print("generator closed")
9
10 g = my_generator()
11
12 print(next(g))
13 print(next(g))
14
15 g.close()
16
17 print(next(g))
```

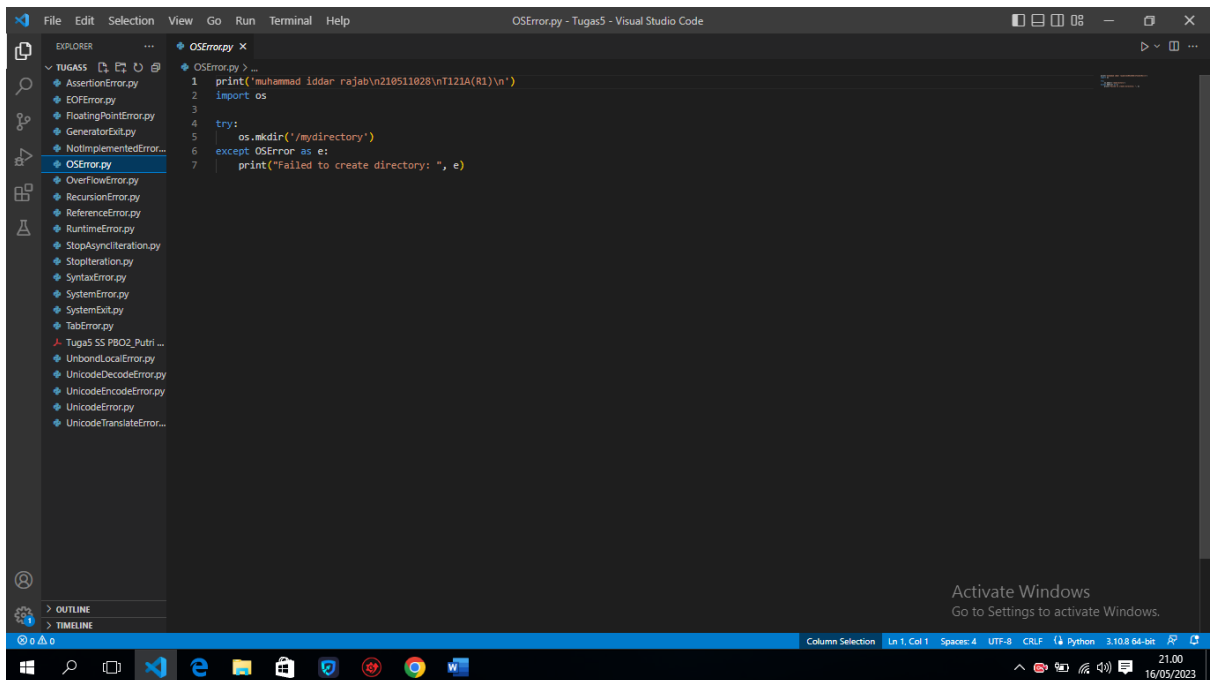
The status bar at the bottom indicates the current position is `Ln 1, Col 1` with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the `Python` interpreter. The system clock shows `20:59` on `16/05/2023`.

5. NotImplemented



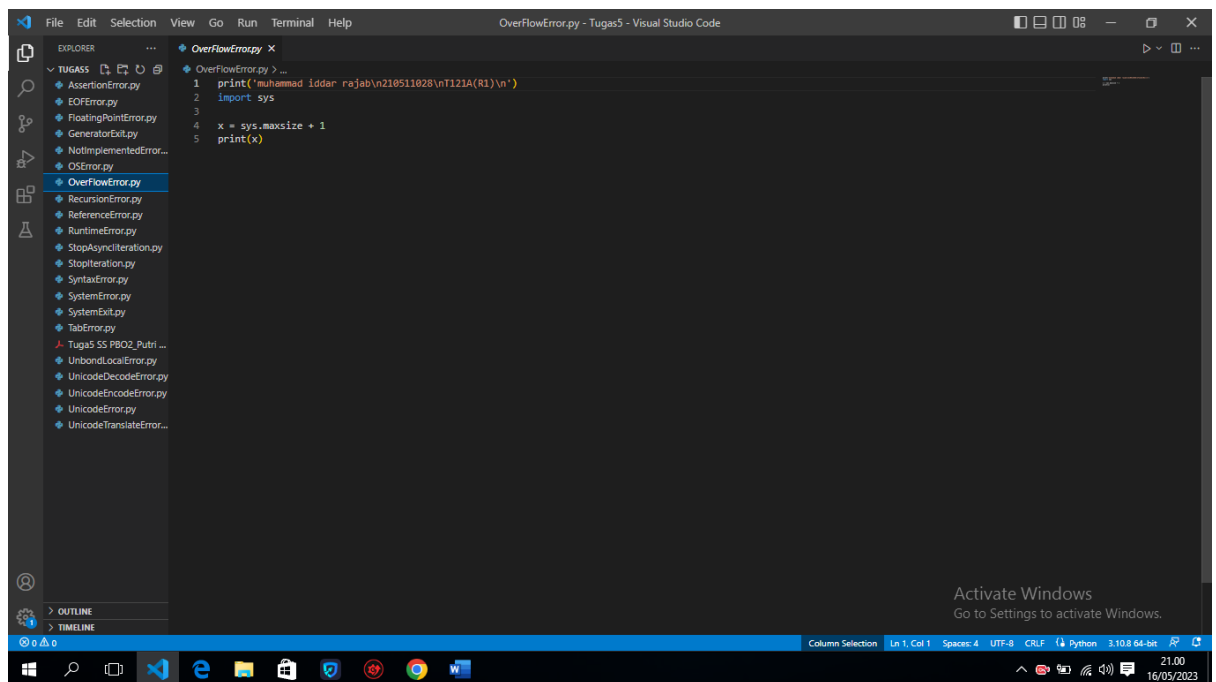
```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 class MyClass:
3     def my_method(self):
4         raise NotImplementedError("Subclasses must override this method")
5
6 my_object = MyClass()
7 my_object.my_method()
```

6. OSError



```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 import os
3
4 try:
5     os.mkdir('/mydirectory')
6 except OSError as e:
7     print('Failed to create directory: ', e)
```

7. OverflowError

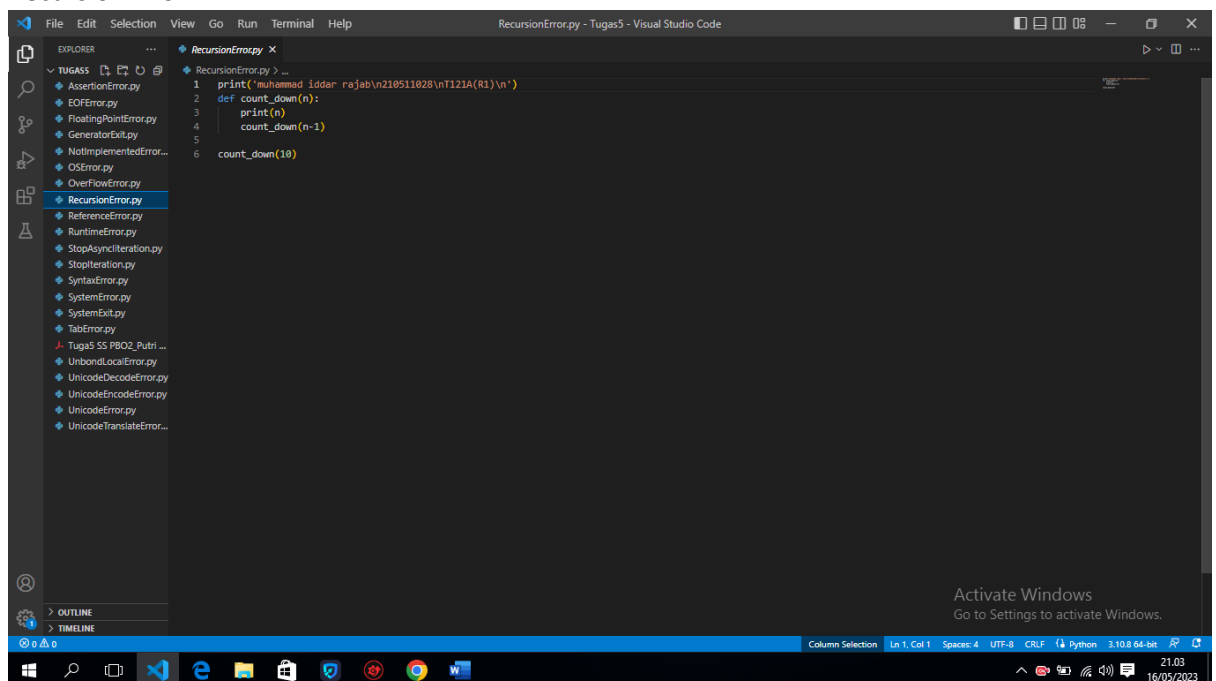


Visual Studio Code interface showing a file named `OverflowError.py`. The code in the editor is:

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 import sys
3
4 x = sys.maxsize + 1
5 print(x)
```

The Explorer sidebar on the left shows a list of files under the `TUGAS5` folder, including `OverflowError.py`, `RecursionError.py`, `ReferenceError.py`, `RuntimeError.py`, `StopAsyncIteration.py`, `StopIteration.py`, `SyntaxError.py`, `SystemError.py`, `SystemExit.py`, `TabError.py`, `Tuga5 SS PBO2_Putri ...`, `UnbondLocalError.py`, `UnicodeDecodeError.py`, `UnicodeEncodeError.py`, `UnicodeError.py`, and `UnicodeTranslateError...`.

8. RecursionError

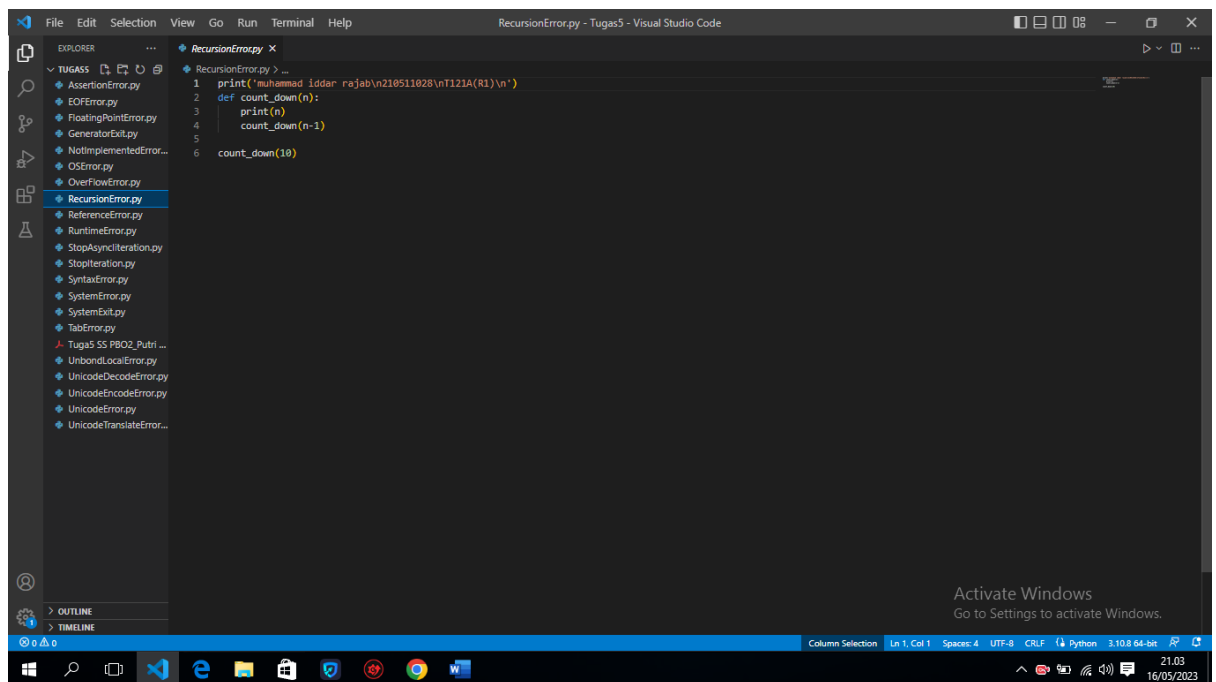


Visual Studio Code interface showing a file named `RecursionError.py`. The code in the editor is:

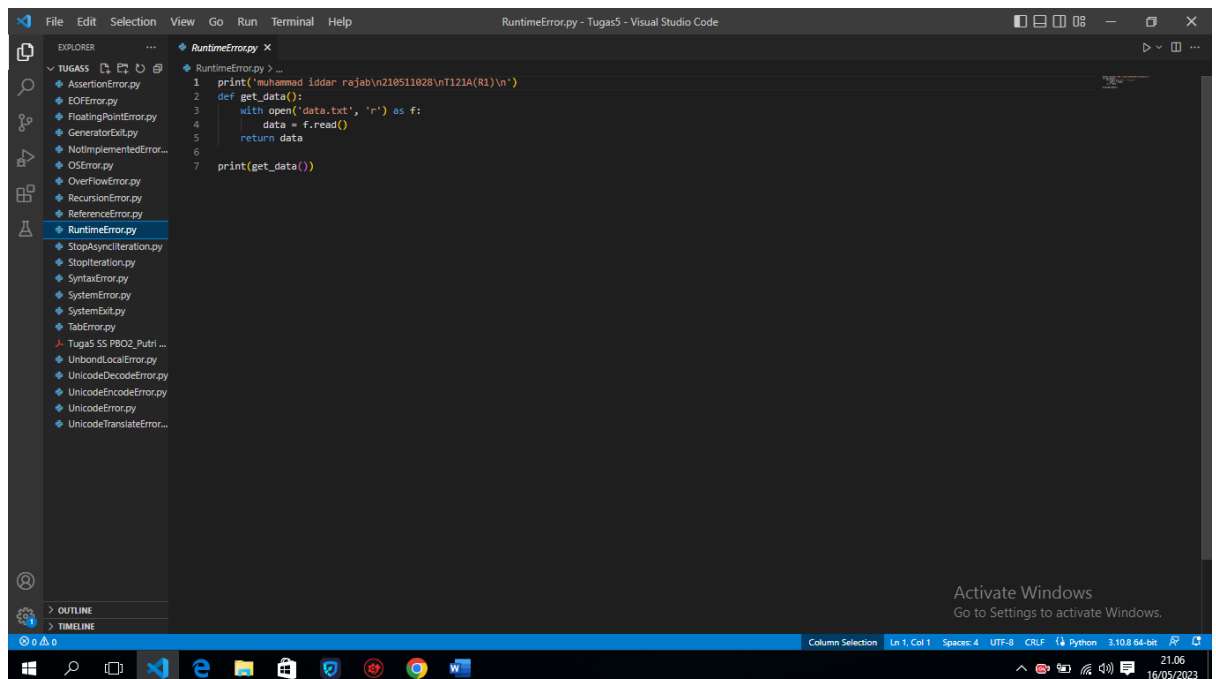
```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 def count_down(n):
3     print(n)
4     count_down(n-1)
5
6 count_down(10)
```

The Explorer sidebar on the left shows a list of files under the `TUGAS5` folder, including `RecursionError.py`, `ReferenceError.py`, `RuntimeError.py`, `StopAsyncIteration.py`, `StopIteration.py`, `SyntaxError.py`, `SystemError.py`, `SystemExit.py`, `TabError.py`, `Tuga5 SS PBO2_Putri ...`, `UnbondLocalError.py`, `UnicodeDecodeError.py`, `UnicodeEncodeError.py`, `UnicodeError.py`, and `UnicodeTranslateError...`.

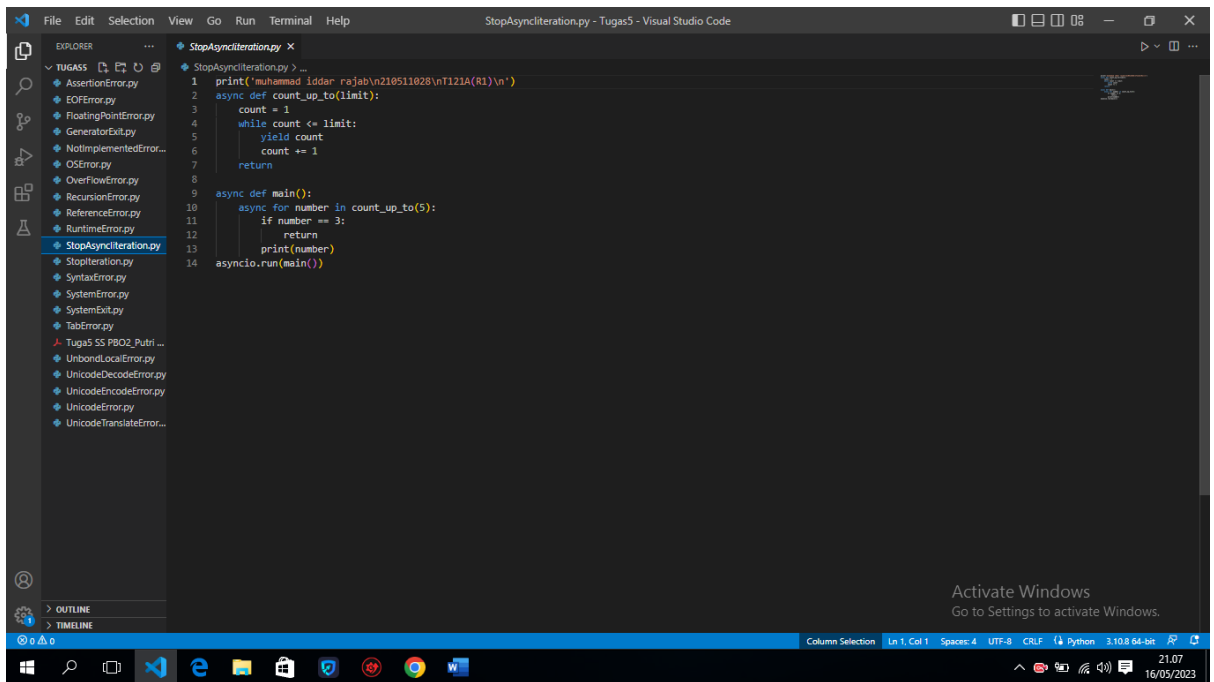
9. ReferenceError



10. RuntimeError

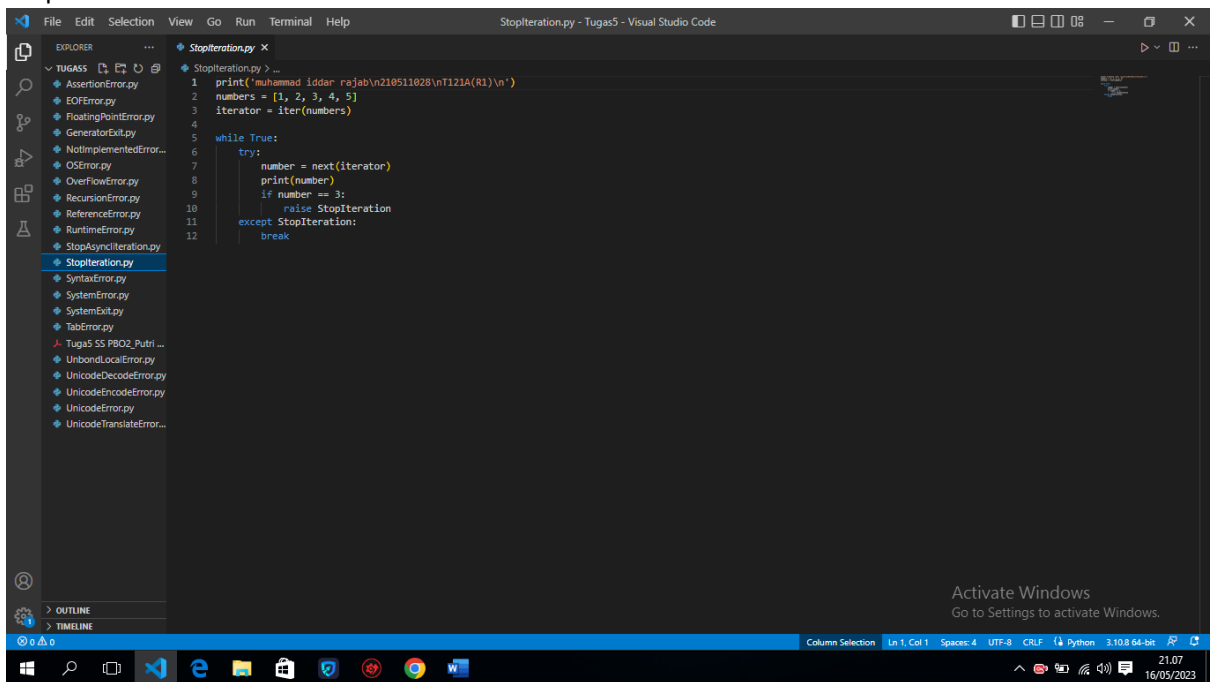


11. StopAsyncIterationError



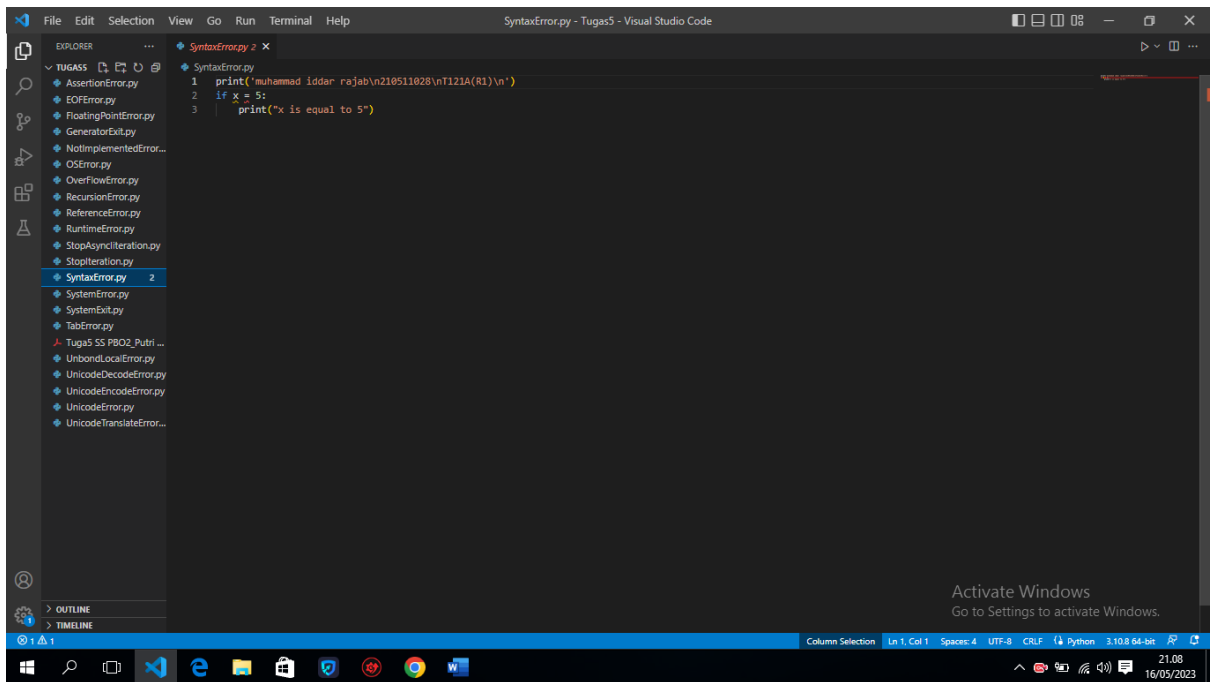
```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 async def count_up_to(limit):
3     count = 1
4     while count <= limit:
5         yield count
6         count += 1
7     return
8
9 async def main():
10     async for number in count_up_to(5):
11         if number == 3:
12             return
13         print(number)
14     asyncio.run(main())
```

12. StopIteration

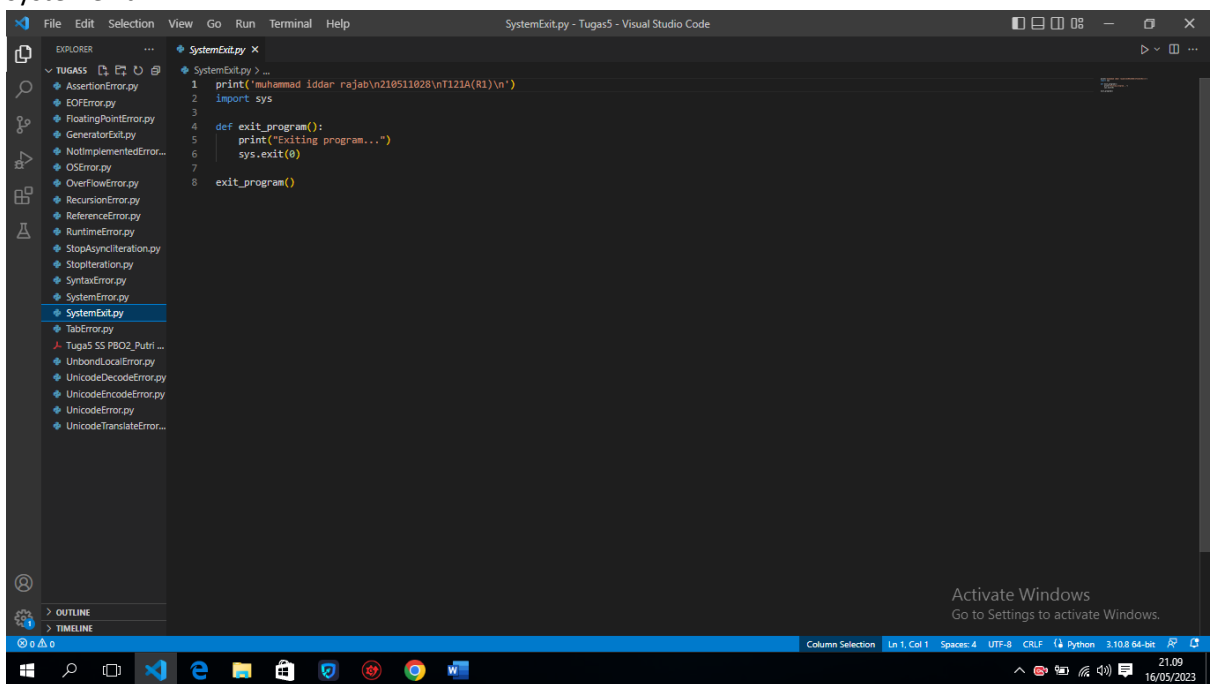


```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 numbers = [1, 2, 3, 4, 5]
3 iterator = iter(numbers)
4
5 while True:
6     try:
7         number = next(iterator)
8         print(number)
9         if number == 3:
10             raise StopIteration
11     except StopIteration:
12         break
```

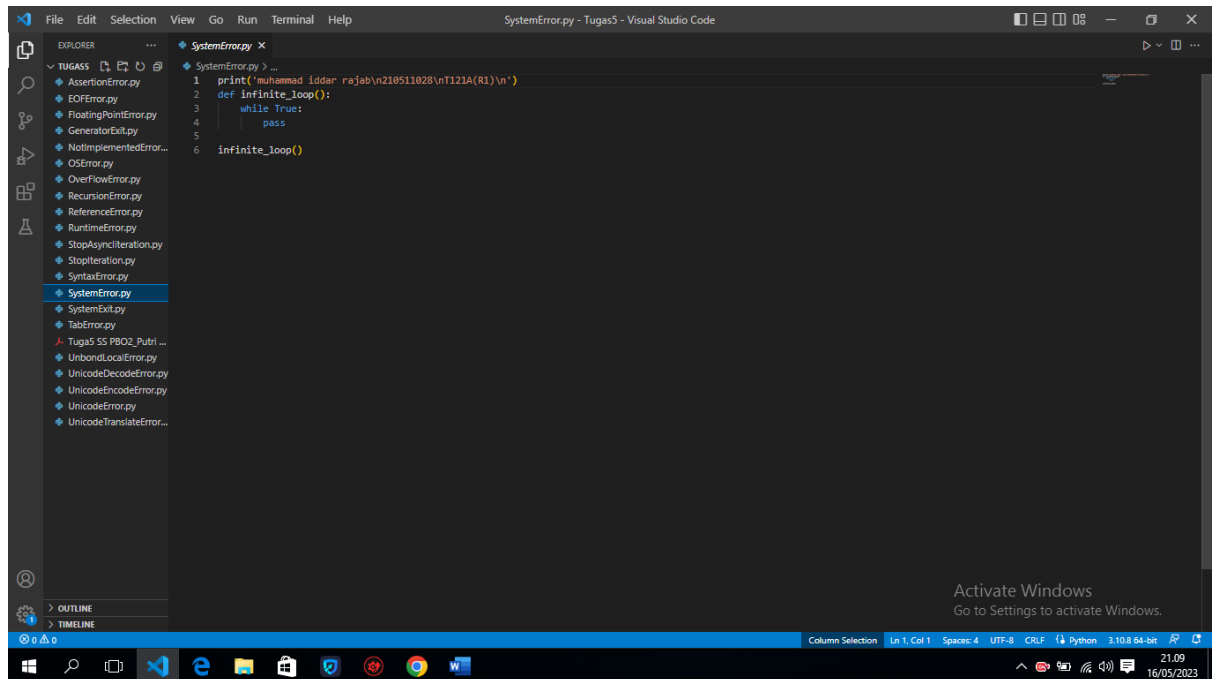
13. syntaxError



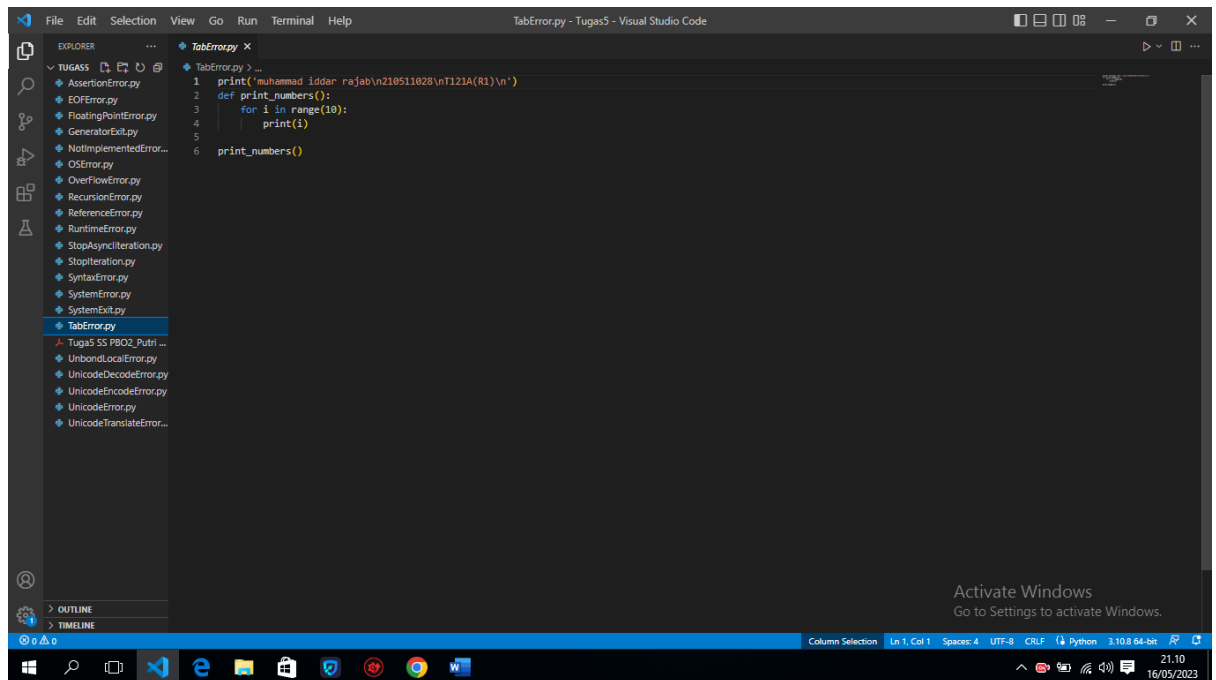
14. systemexit



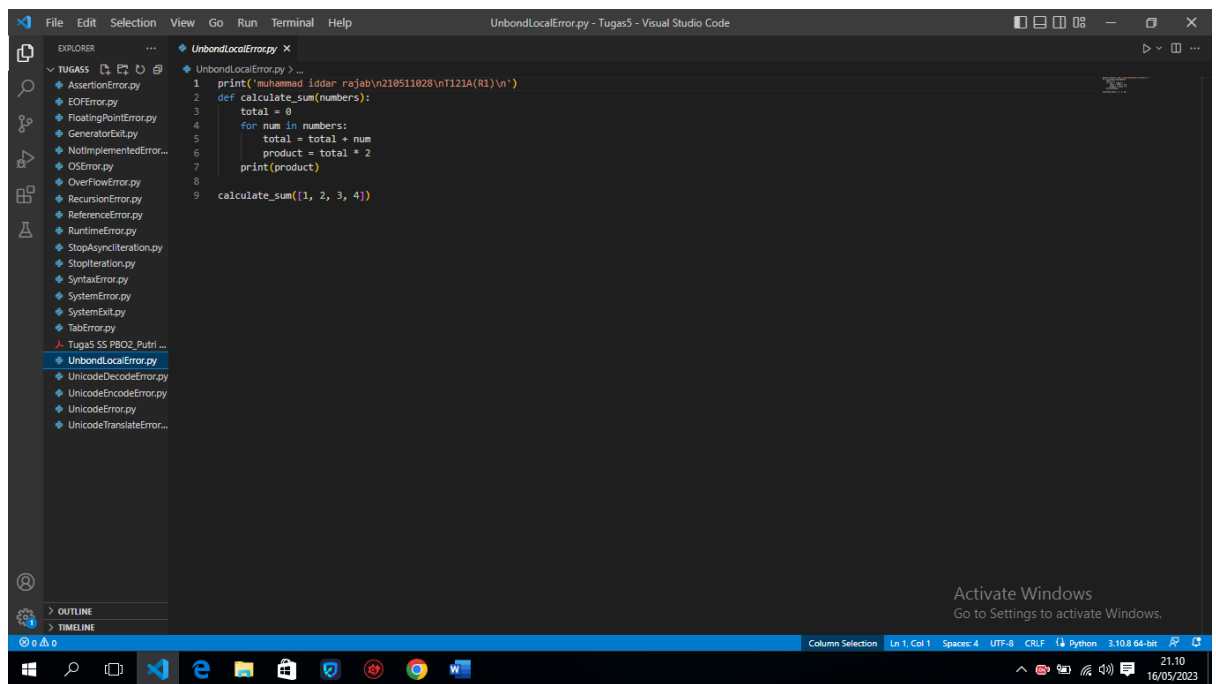
15.systemError



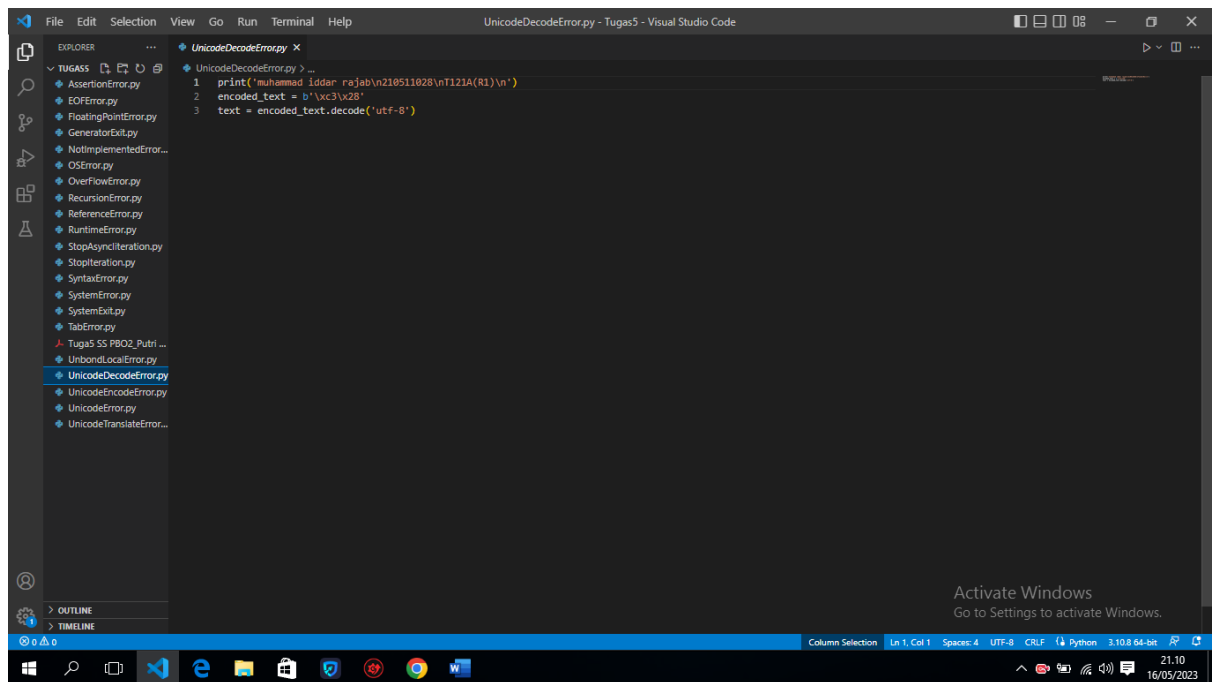
16. TabError



17. Unbondlocal



18. unicodeDecode



19. unicodeEncode

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 text = 'Häagen-Dazs'
3 encoded_text = text.encode('ascii')
```

20. unicodeError

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2 text = "Hello, world! \u00ff"
3 print(text)
```

21. unicodetranlite

