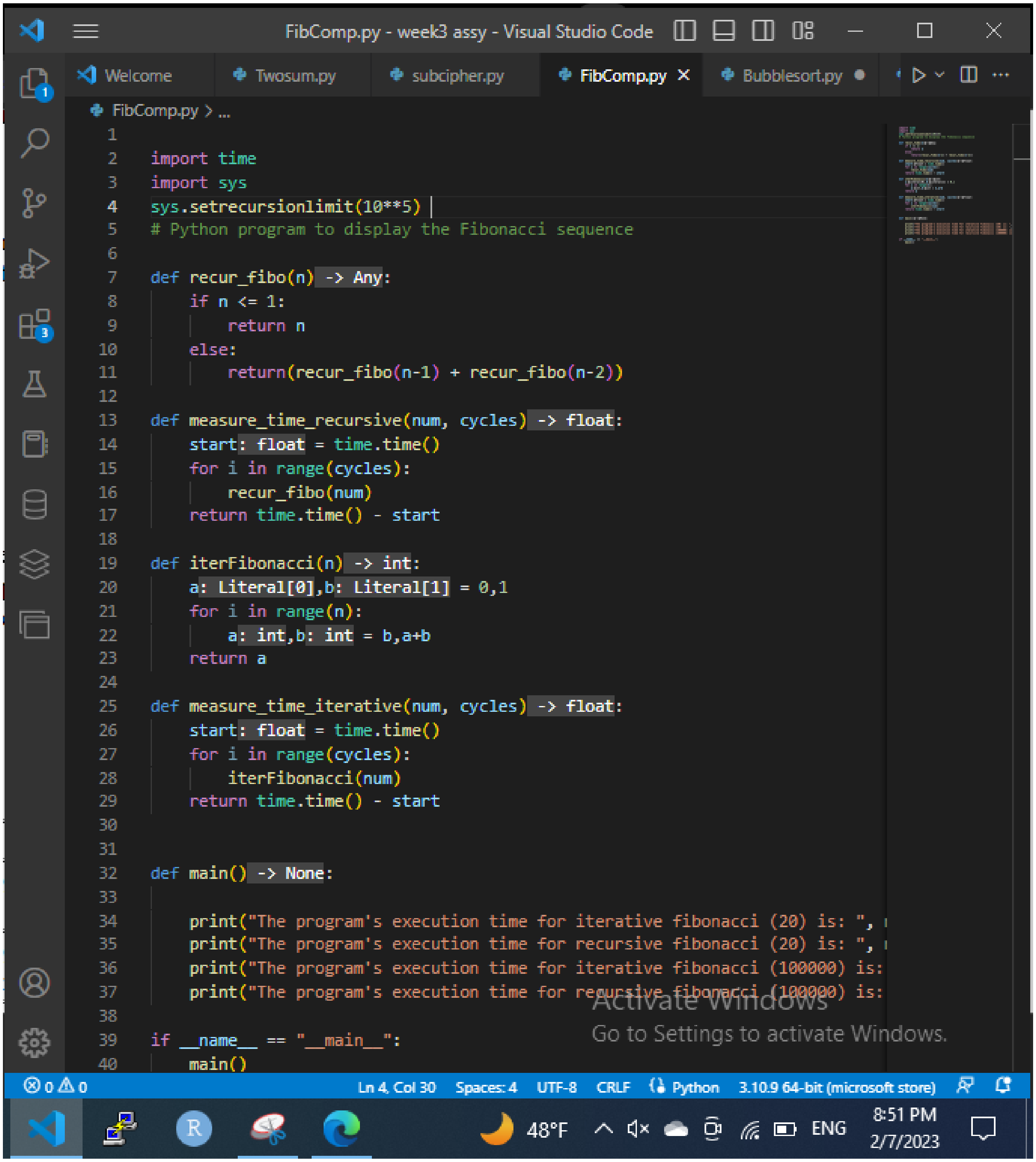
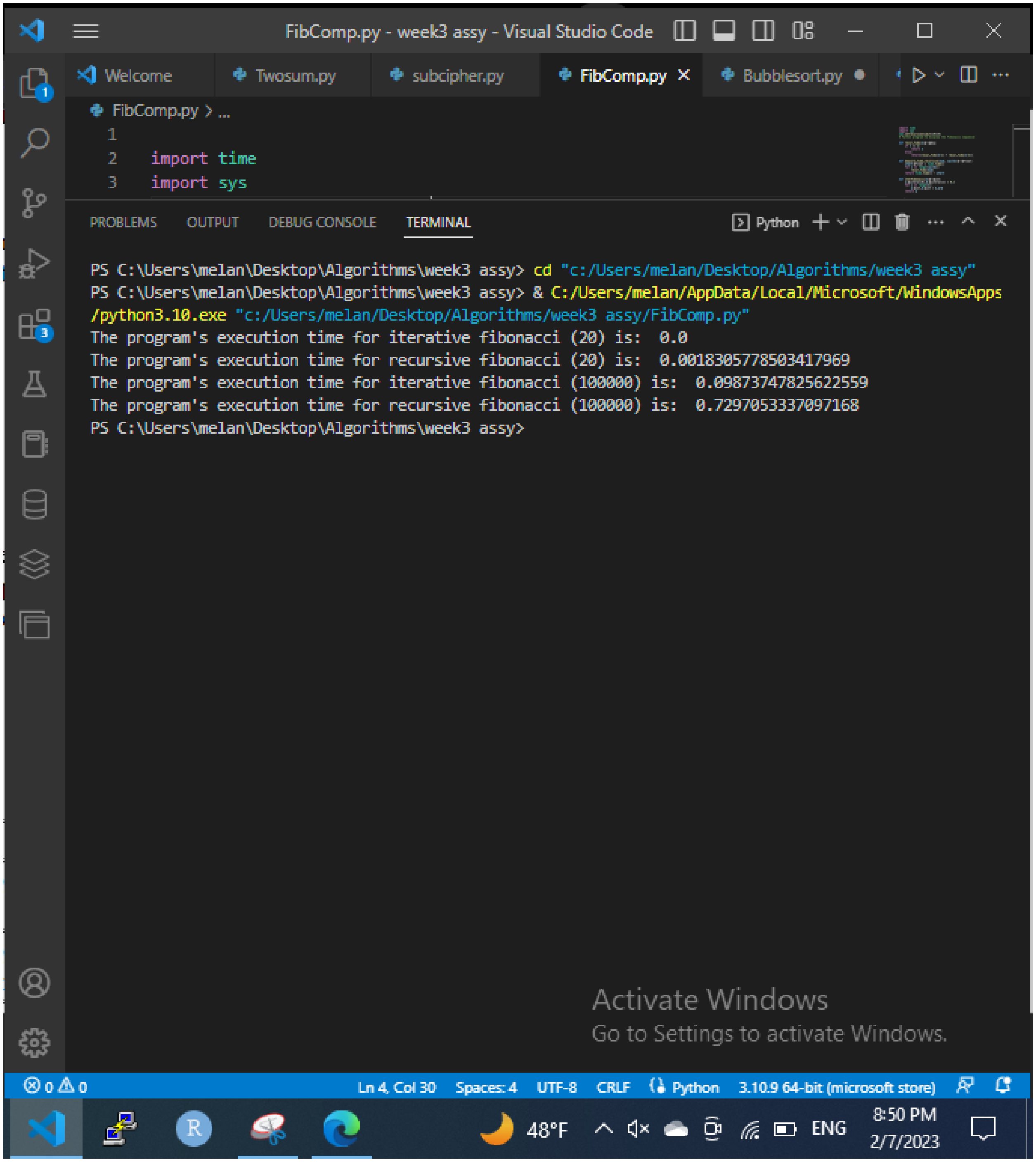


```python

import time
import sys
sys.setrecursionlimit(10**5) 
# Python program to display the Fibonacci sequence

def recur_fibo(n) -> Any:
    if n <= 1:
        return n
    else:
        return(recur_fibo(n-1) + recur_fibo(n-2))

def measure_time_recursive(num, cycles) -> float:
    start: float = time.time()
    for i in range(cycles):
        recur_fibo(num)
    return time.time() - start

def iterFibonacci(n) -> int:
    a: Literal[0],b: Literal[1] = 0,1
    for i in range(n):
        a: int,b: int = b,a+b
    return a

def measure_time_iterative(num, cycles) -> float:
    start: float = time.time()
    for i in range(cycles):
        iterFibonacci(num)
    return time.time() - start


def main() -> None:

    print("The program's execution time for iterative fibonacci (20) is: ",
    print("The program's execution time for recursive fibonacci (20) is: ",
    print("The program's execution time for iterative fibonacci (100000) is:
    print("The program's execution time for recursive fibonacci (100000) is:

if __name__ == "__main__":
    main()
```



```python

import time
import sys
```

```
PS C:\Users\melan\Desktop\Algorithms\week3 assy> cd "c:/Users/melan/Desktop/Algorithms/week3 assy"
PS C:\Users\melan\Desktop\Algorithms\week3 assy> & C:/Users/melan/AppData/Local/Microsoft/WindowsApps
/python3.10.exe "c:/Users/melan/Desktop/Algorithms/week3 assy/FibComp.py"
The program's execution time for iterative fibonacci (20) is:  0.0
The program's execution time for recursive fibonacci (20) is:  0.0018305778503417969
The program's execution time for iterative fibonacci (100000) is:  0.0987374782562259
The program's execution time for recursive fibonacci (100000) is:  0.7297053337097168
PS C:\Users\melan\Desktop\Algorithms\week3 assy>
```

CODE

```
import time
import sys
sys.setrecursionlimit(10**5)
# Python program to display the Fibonacci sequence

def recur_fibo(n):
    if n <= 1:
        return n
    else:
        return(recur_fibo(n-1) + recur_fibo(n-2))

def measure_time_recursive(num, cycles):
    start = time.time()
    for i in range(cycles):
        recur_fibo(num)
    return time.time() - start

def iterFibonacci(n):
    a,b = 0,1
    for i in range(n):
        a,b = b,a+b
    return a

def measure_time_iterative(num, cycles):
    start = time.time()
    for i in range(cycles):
        iterFibonacci(num)
    return time.time() - start


def main():

    print("The program's execution time for iterative fibonacci (20) is: ",
measure_time_iterative(7, 20))
    print("The program's execution time for recursive fibonacci (20) is: ",
measure_time_recursive(7, 20))
    print("The program's execution time for iterative fibonacci (100000) is: ",
measure_time_iterative(7, 100000))
    print("The program's execution time for recursive fibonacci (100000) is: ",
measure_time_recursive(7, 100000))
```

```
if __name__ == "__main__":
    main()
```