

```
fibNumbers.py > ...
1 def fibonacci(n):
2     '''Calculate the fibonacci number of n'''
3
4     # Base case
5     if n == 0 or n == 1:
6         return n
7
8     return fibonacci(n - 1) + fibonacci(n - 2)
9
10 for i in range(10):
11     print(f"F({i}) = ", fibonacci(i))
12
13 # Output:
14 fibonacci(2**10)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```

File "c:\Users\matt\OneDrive\CityU\2024-04 Fall\CS-469 Data Structure\HOS\cs469-hos03-fall-2024-Matt-PMCT\fibNumbers.py", line 8, in fibonacci
    return fibonacci(n - 1) + fibonacci(n - 2)
File "c:\Users\matt\OneDrive\CityU\2024-04 Fall\CS-469 Data Structure\HOS\cs469-hos03-fall-2024-Matt-PMCT\fibNumbers.py", line 8, in fibonacci
    return fibonacci(n - 1) + fibonacci(n - 2)
[Previous line repeated 996 more times]
RecursionError: maximum recursion depth exceeded
PS C:\Users\matt\OneDrive\CityU\2024-04 Fall\CS-469 Data Structure\HOS\cs469-hos03-fall-2024-Matt-PMCT> 
```

The error is “maximum recursion depth exceeded” is due to the default Python recursion depth limit of 1000 ([reference](#)). There is ability to change that reference limit, but it is in place to reduce the likelihood of freezing the machine or crashing it due to extraordinary recursion. To offer a better user experience I could introduce error handling to catch the error and ask the user to input a smaller number, or just check the number size prior to attempting to execute it and provide a warning.