# ASSIGNMENT - 5.5

**HT.NO: 2303A510I4**

**BATCH: 30**

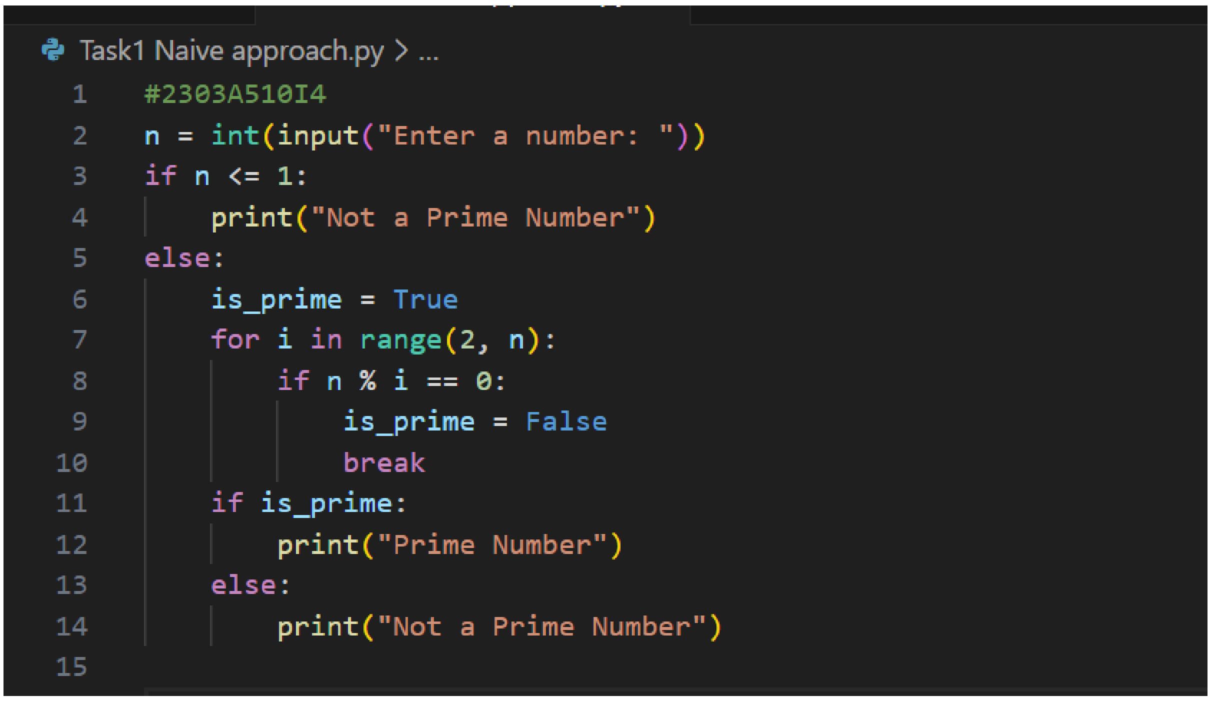
## TASK 1: Transparency in Algorithm Optimization

**Prompt:**

Generate Python code for two prime-checking methods and explain how the optimized version improves performance.

**CODE:**

**Naive approach(basic)**

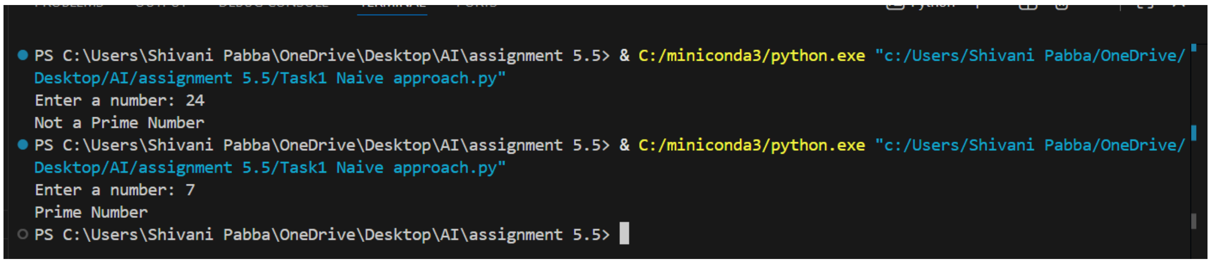


Task1 Naive approach.py > ...

```python
#2303A510I4
n = int(input("Enter a number: "))
if n <= 1:
    print("Not a Prime Number")
else:
    is_prime = True
    for i in range(2, n):
        if n % i == 0:
            is_prime = False
            break
    if is_prime:
        print("Prime Number")
    else:
        print("Not a Prime Number")

```

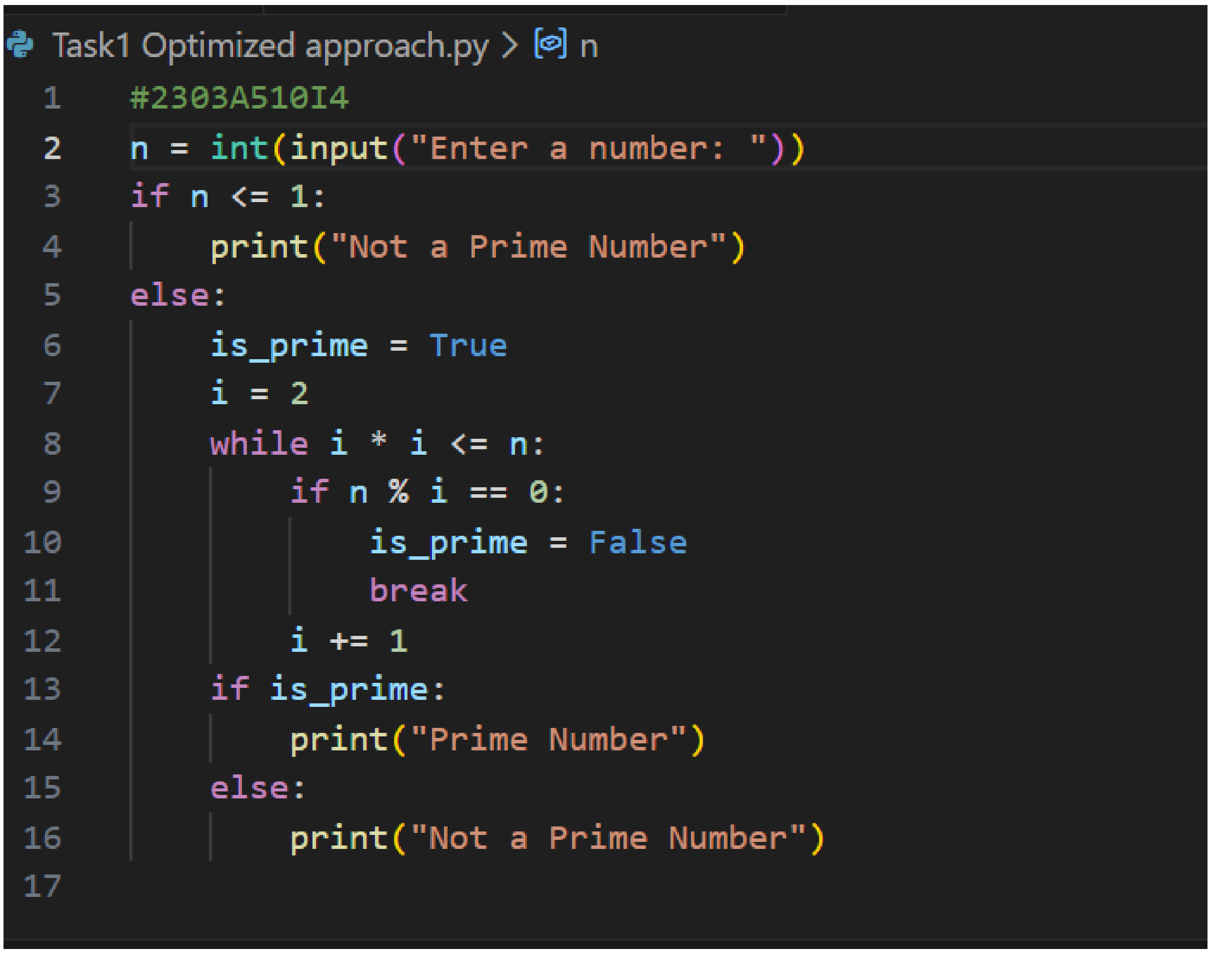
**OUTPUT:**



```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI\assignment 5.5> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 5.5/Task1 Naive approach.py"
Enter a number: 24
Not a Prime Number
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI\assignment 5.5> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 5.5/Task1 Naive approach.py"
Enter a number: 7
Prime Number
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI\assignment 5.5>
```

**CODE:**

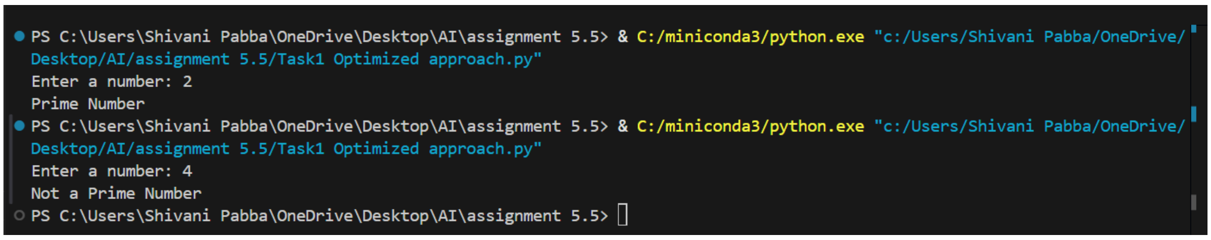
**Optimized approach**


Task1 Optimized approach.py > n

```python
#2303A510I4
n = int(input("Enter a number: "))
if n <= 1:
    print("Not a Prime Number")
else:
    is_prime = True
    i = 2
    while i * i <= n:
        if n % i == 0:
            is_prime = False
            break
        i += 1
    if is_prime:
        print("Prime Number")
    else:
        print("Not a Prime Number")

```

**OUTPUT:**


```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI\assignment 5.5> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/
Desktop/AI/assignment 5.5/Task1 Optimized approach.py"
Enter a number: 2
Prime Number
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI\assignment 5.5> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/
Desktop/AI/assignment 5.5/Task1 Optimized approach.py"
Enter a number: 4
Not a Prime Number
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI\assignment 5.5> 
```

**Observation:**

**Naive Method**

This method checks almost every number to see if it divides the given value, so it takes more time as the number increases.

## Optimized Method

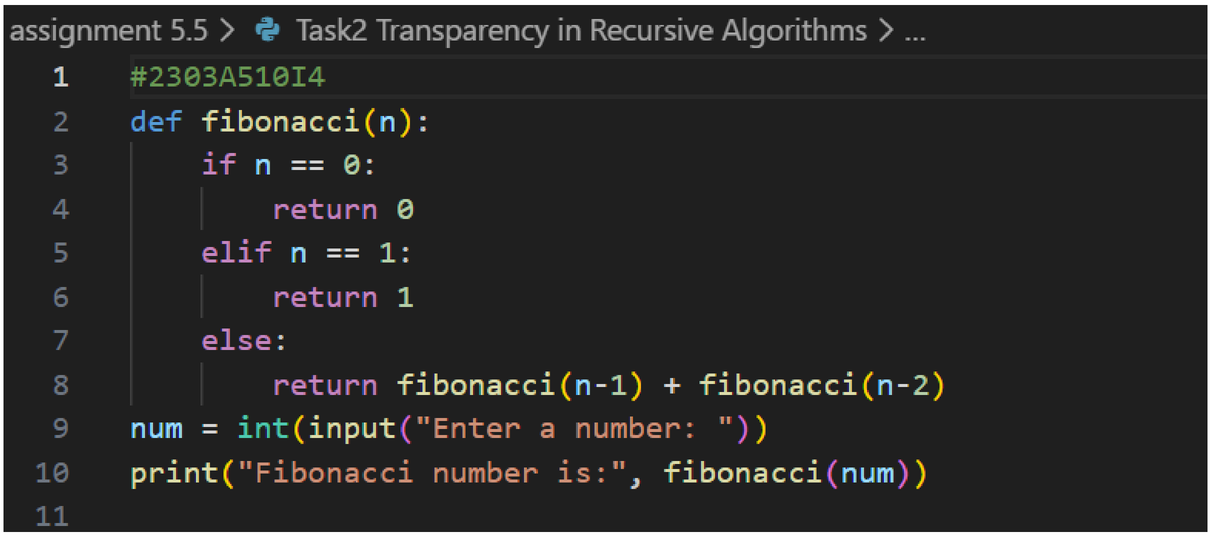
This method checks only the required divisors up to the square root, so it gives the result much faster.

## TASK 2: Transparency in Recursive Algorithms

### Prompt:

Generate a recursive Python function to calculate Fibonacci numbers. Add clear comments explaining how recursion works, including base cases and recursive calls.

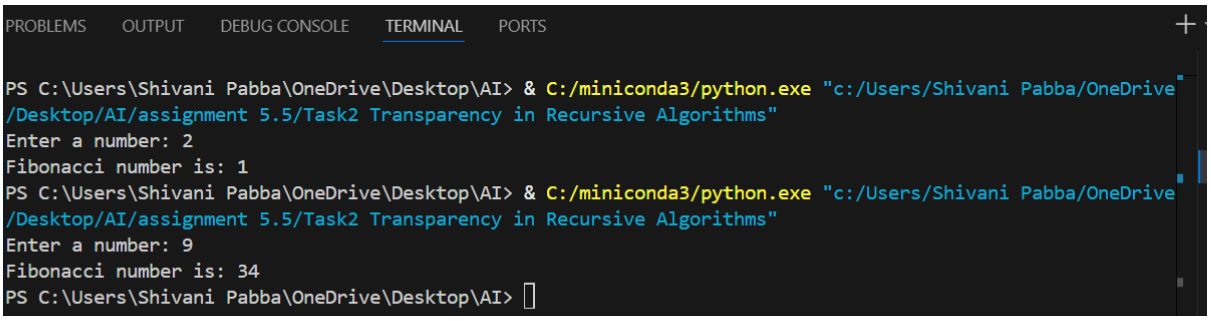
### Code:



assignment 5.5 > Task2 Transparency in Recursive Algorithms > ...

```python
#2303A510I4
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)
num = int(input("Enter a number: "))
print("Fibonacci number is:", fibonacci(num))
```

### Output:



```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive
/Desktop/AI/assignment 5.5/Task2 Transparency in Recursive Algorithms"
Enter a number: 2
Fibonacci number is: 1
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive
/Desktop/AI/assignment 5.5/Task2 Transparency in Recursive Algorithms"
Enter a number: 9
Fibonacci number is: 34
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> 
```

**Observation:**

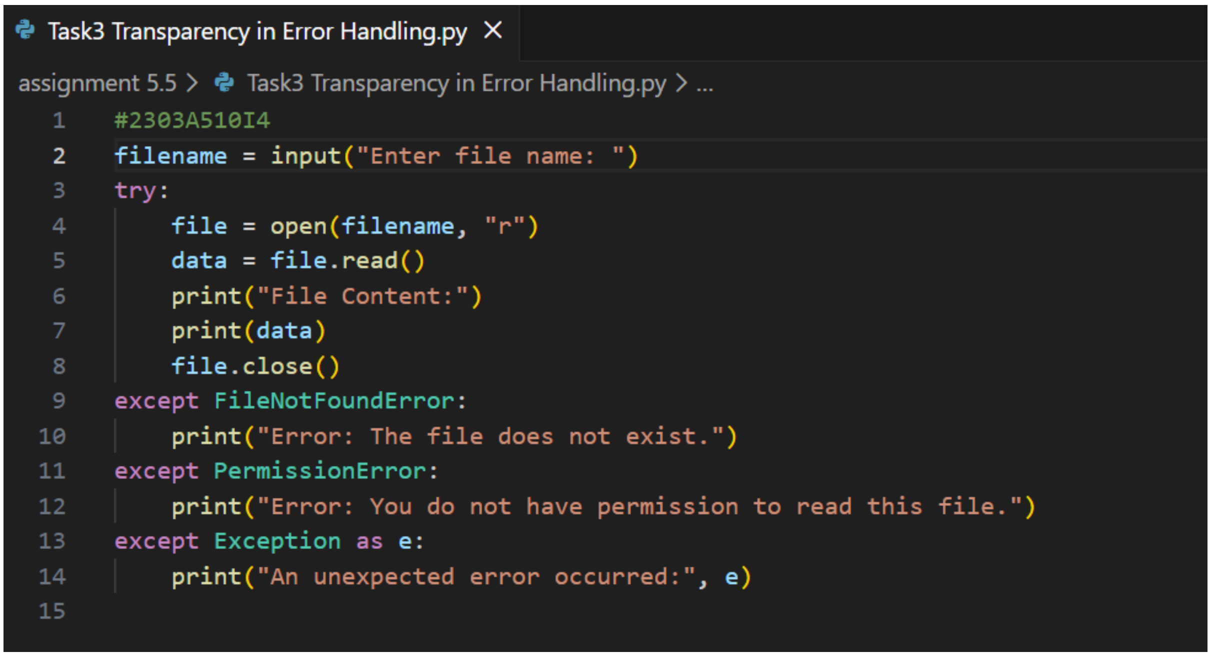
The recursive function breaks the problem into smaller parts and solves them step by step until it reaches a stopping point, after which the final result is obtained.

**TASK 3: Transparency in Error Handling**

**Prompt:**

Generate code with proper error handling and clear explanations for each exception.

**Code:**

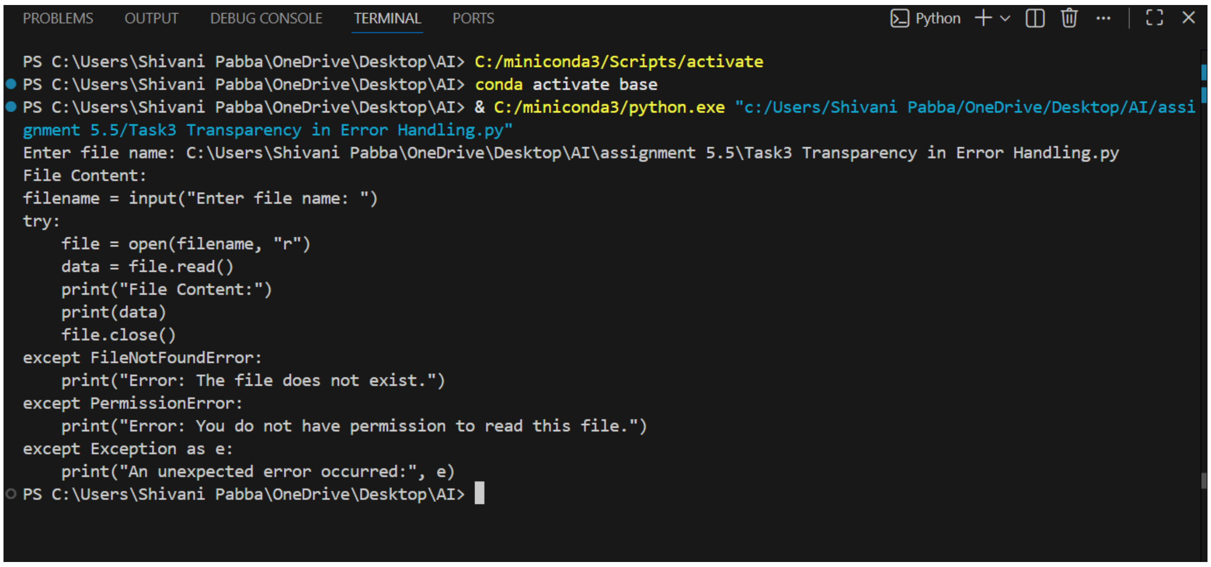


Task3 Transparency in Error Handling.py

assignment 5.5 > Task3 Transparency in Error Handling.py > ...

```python
#2303A510I4
filename = input("Enter file name: ")
try:
    file = open(filename, "r")
    data = file.read()
    print("File Content:")
    print(data)
    file.close()
except FileNotFoundError:
    print("Error: The file does not exist.")
except PermissionError:
    print("Error: You do not have permission to read this file.")
except Exception as e:
    print("An unexpected error occurred:", e)
```

## Output:



```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assi
gnment 5.5/Task3 Transparency in Error Handling.py"
Enter file name: C:\Users\Shivani Pabba\OneDrive\Desktop\AI\assignment 5.5\Task3 Transparency in Error Handling.py
File Content:
filename = input("Enter file name: ")
try:
    file = open(filename, "r")
    data = file.read()
    print("File Content:")
    print(data)
    file.close()
except FileNotFoundError:
    print("Error: The file does not exist.")
except PermissionError:
    print("Error: You do not have permission to read this file.")
except Exception as e:
    print("An unexpected error occurred:", e)
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI>
```

## Observation:

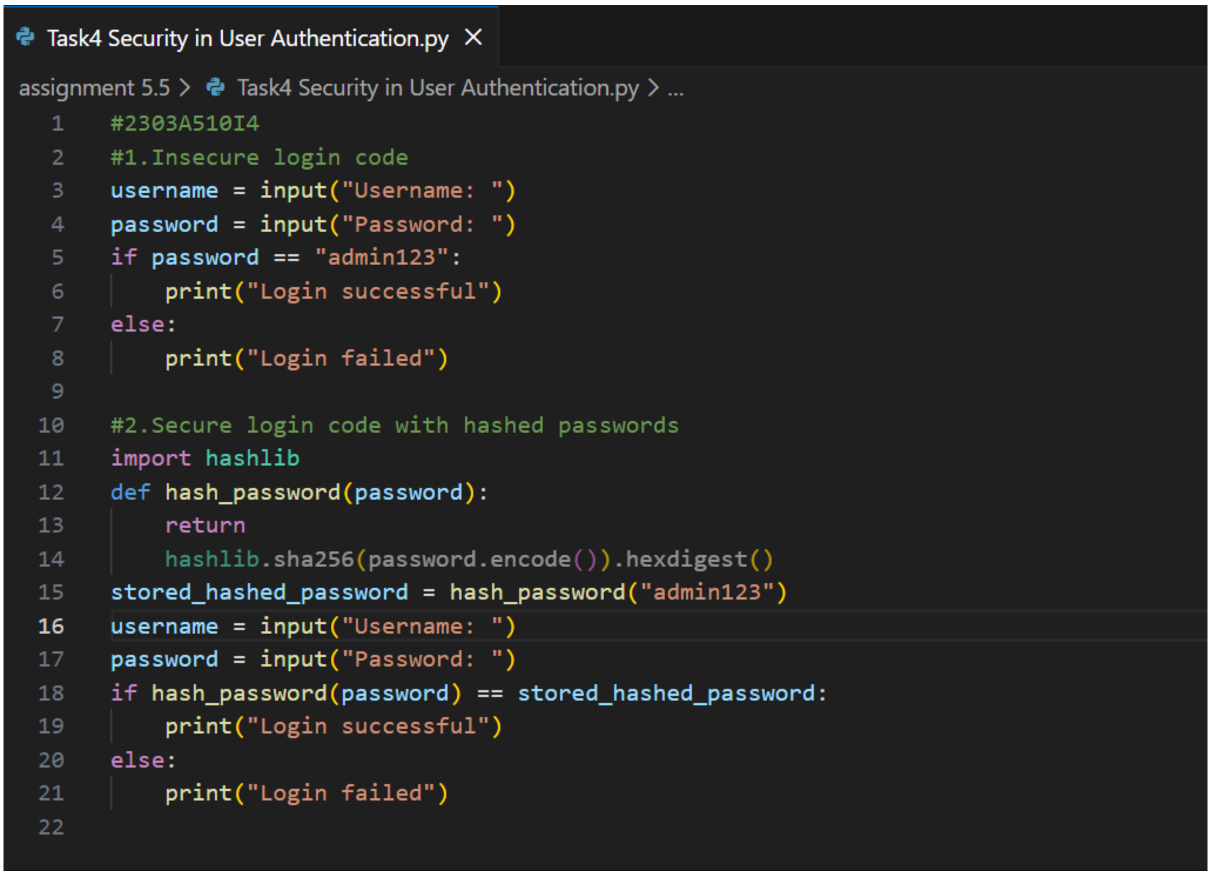
The program safely reads the file and clearly reports errors when the file is missing or inaccessible, instead of stopping unexpectedly.

## TASK 4: Security in User Authentication

## Prompt:

Generate a simple Python-based login system.

## Code:



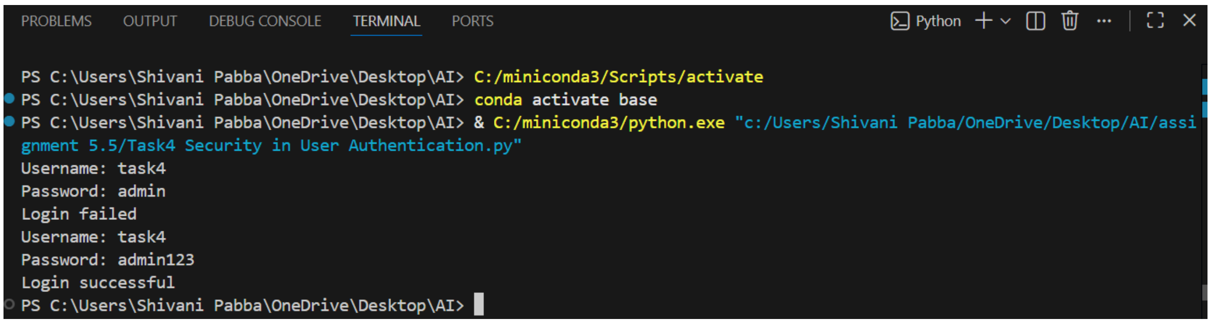
Task4 Security in User Authentication.py
assignment 5.5 > Task4 Security in User Authentication.py > ...

```python
#2303A510I4
#1.Insecure login code
username = input("Username: ")
password = input("Password: ")
if password == "admin123":
    print("Login successful")
else:
    print("Login failed")

#2.Secure login code with hashed passwords
import hashlib
def hash_password(password):
    return
    hashlib.sha256(password.encode()).hexdigest()
stored_hashed_password = hash_password("admin123")
username = input("Username: ")
password = input("Password: ")
if hash_password(password) == stored_hashed_password:
    print("Login successful")
else:
    print("Login failed")

```

## Output:



```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assi
gnment 5.5/Task4 Security in User Authentication.py"
Username: task4
Password: admin
Login failed
Username: task4
Password: admin123
Login successful
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> 
```

## Observation:

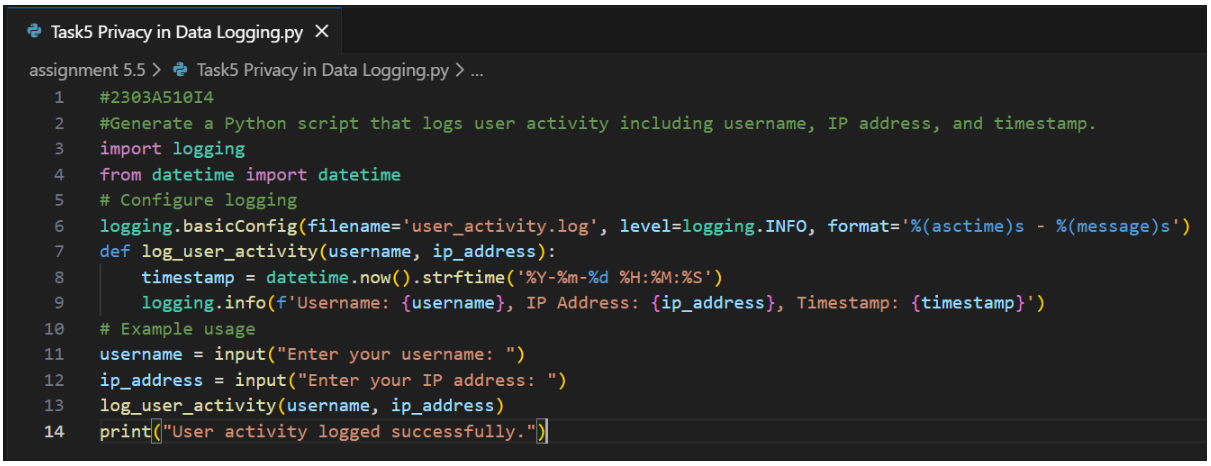
The first code is insecure because it uses plain-text passwords, while the second code improves security by using password hashing.

## TASK 5: Privacy in Data Logging

### Prompt:

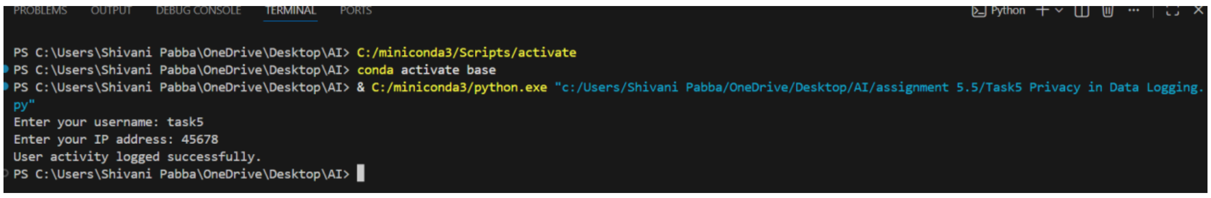
Generate a Python script that logs user activity including username, IP address, and timestamp.

### Code:


Task5 Privacy in Data Logging.py
assignment 5.5 > Task5 Privacy in Data Logging.py > ...

```python
#2303A510I4
#Generate a Python script that logs user activity including username, IP address, and timestamp.
import logging
from datetime import datetime
# Configure logging
logging.basicConfig(filename='user_activity.log', level=logging.INFO, format='%(asctime)s - %(message)s')
def log_user_activity(username, ip_address):
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    logging.info(f'Username: {username}, IP Address: {ip_address}, Timestamp: {timestamp}')
# Example usage
username = input("Enter your username: ")
ip_address = input("Enter your IP address: ")
log_user_activity(username, ip_address)
print("User activity logged successfully.")
```

### Output:



```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 5.5/Task5 Privacy in Data Logging.
py"
Enter your username: task5
Enter your IP address: 45678
User activity logged successfully.
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> 
```

### Observation:

The program logs complete user details such as username and IP address, which may expose sensitive information and create privacy risks if the log file is accessed without proper security.