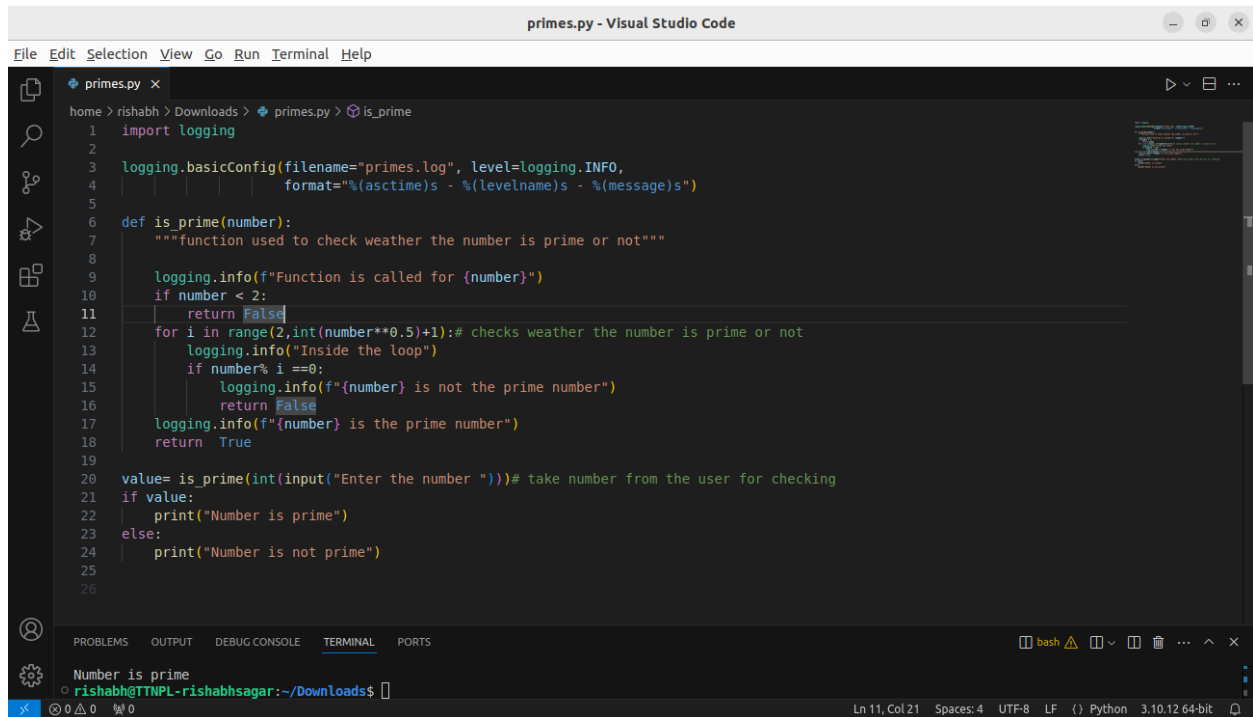


Introduction to Python

Q1)Write a Python script to test if a number is prime or not? - The Script name: primes.py
- Add a functions is_prime() which return boolean True or False - Program should accept a number from console

Ans.

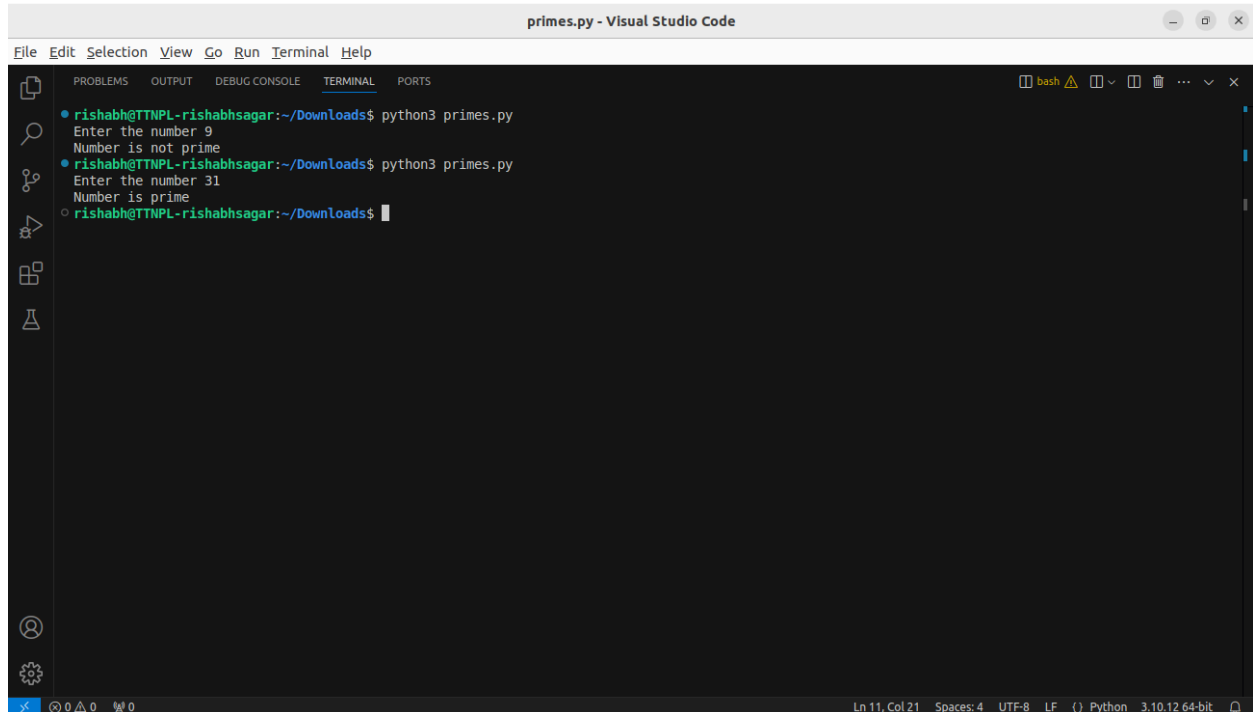


The screenshot shows the Visual Studio Code editor with a file named 'primes.py' open. The code defines a function 'is_prime' that checks if a number is prime. It uses logging to track function calls and loop iterations. The function returns True for prime numbers and False for non-prime numbers. The script prompts the user to enter a number and prints the result.

```
1 import logging
2
3 logging.basicConfig(filename="primes.log", level=logging.INFO,
4                     format="%(asctime)s - %(levelname)s - %(message)s")
5
6 def is_prime(number):
7     """function used to check weather the number is prime or not"""
8
9     logging.info(f"Function is called for {number}")
10    if number < 2:
11        return False
12    for i in range(2,int(number**0.5)+1):# checks weather the number is prime or not
13        logging.info("Inside the loop")
14        if number%i ==0:
15            logging.info(f"{number} is not the prime number")
16            return False
17        logging.info(f"{number} is the prime number")
18    return True
19
20 value= is_prime(int(input("Enter the number ")))# take number from the user for checking
21 if value:
22     print("Number is prime")
23 else:
24     print("Number is not prime")
25
26
```

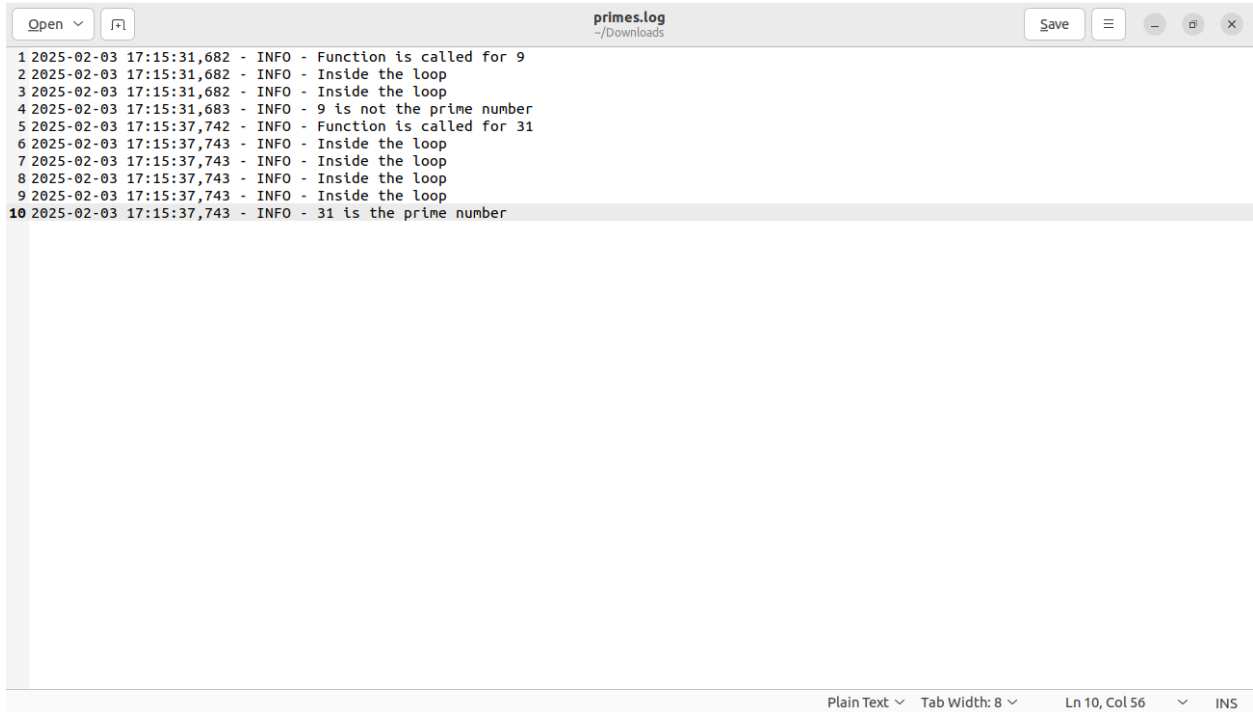
The terminal output shows the execution of the script:

```
Number is prime
rishabh@TTNPL-rishabsagar:~/Downloads$
```



The screenshot shows the terminal output of the script. It demonstrates two test cases: one for the number 9, which is not prime, and one for the number 31, which is prime.

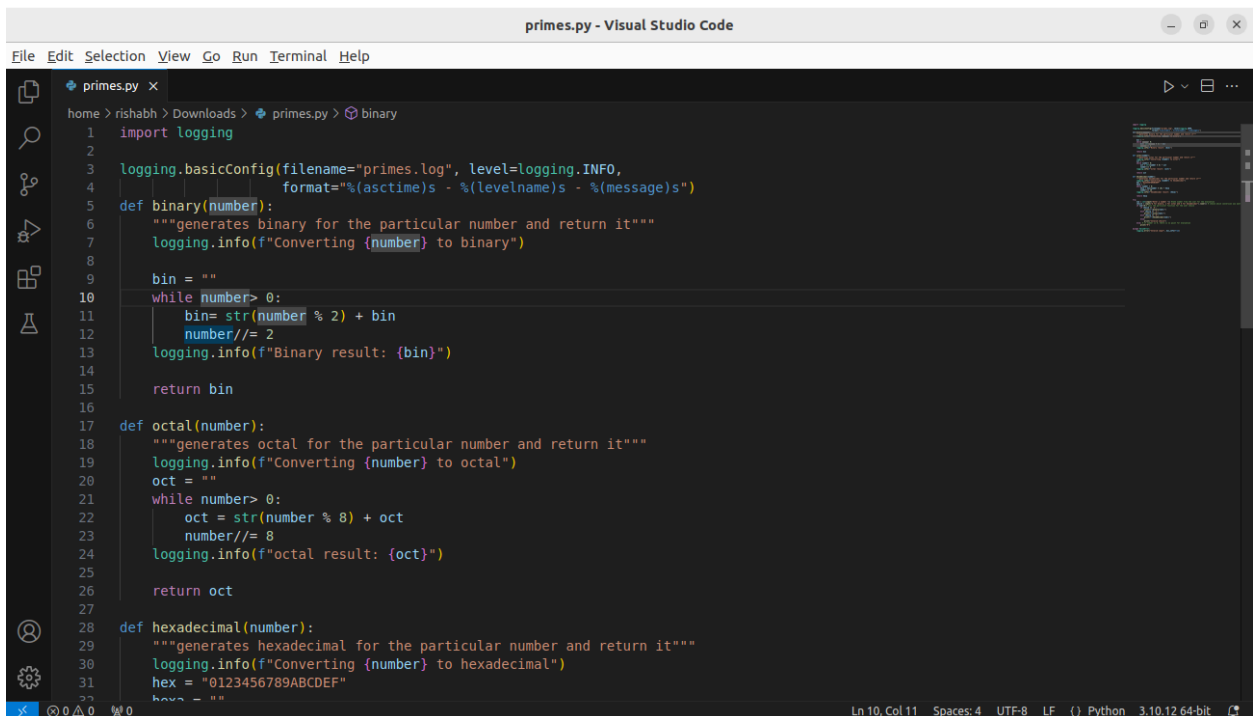
```
rishabh@TTNPL-rishabsagar:~/Downloads$ python3 primes.py
Enter the number 9
Number is not prime
rishabh@TTNPL-rishabsagar:~/Downloads$ python3 primes.py
Enter the number 31
Number is prime
rishabh@TTNPL-rishabsagar:~/Downloads$
```



```
1 2025-02-03 17:15:31,682 - INFO - Function is called for 9
2 2025-02-03 17:15:31,682 - INFO - Inside the loop
3 2025-02-03 17:15:31,682 - INFO - Inside the loop
4 2025-02-03 17:15:31,683 - INFO - 9 is not the prime number
5 2025-02-03 17:15:37,742 - INFO - Function is called for 31
6 2025-02-03 17:15:37,743 - INFO - Inside the loop
7 2025-02-03 17:15:37,743 - INFO - Inside the loop
8 2025-02-03 17:15:37,743 - INFO - Inside the loop
9 2025-02-03 17:15:37,743 - INFO - Inside the loop
10 2025-02-03 17:15:37,743 - INFO - 31 is the prime number
```

Q2)Write a code to print binary, octal or hexa-decimal presentation of a number. Do not use any third party library.

Ans.



```
primes.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help

1 import logging
2
3 logging.basicConfig(filename="primes.log", level=logging.INFO,
4                     format="%asctime)s - %(levelname)s - %(message)s")
5
6 def binary(number):
7     """generates binary for the particular number and return it"""
8     logging.info(f"Converting {number} to binary")
9
10    bin = ""
11    while number > 0:
12        bin = str(number % 2) + bin
13        number //= 2
14    logging.info(f"Binary result: {bin}")
15
16    return bin
17
18 def octal(number):
19     """generates octal for the particular number and return it"""
20     logging.info(f"Converting {number} to octal")
21     oct = ""
22     while number > 0:
23         oct = str(number % 8) + oct
24         number //= 8
25     logging.info(f"octal result: {oct}")
26
27     return oct
28
29 def hexadecimal(number):
30     """generates hexadecimal for the particular number and return it"""
31     logging.info(f"Converting {number} to hexadecimal")
32     hex = "0123456789ABCDEF"
33     hexa = ""
```

primes.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

primes.py x

```
home > rishabh > Downloads > primes.py > binary
28 def hexadecimal(number):
29     """
30     generates hexadecimal for the particular number and return it
31     """
32     logging.info(f"Converting {number} to hexadecimal")
33     hex = "0123456789ABCDEF"
34     hexa = ""
35     while number > 0:
36         hexa = hex[number % 16] + hexa
37         number //= 16
38     logging.info(f"hexadecimal result: {hexa}")
39
40     return hexa
41
42 try:
43     num = int(input("Enter a number ")) #take number from the user for the evaluation
44     choice = input("b for binary, o for octal and h for hexadecimal").lower() # choose which conversion you wanted
45     if num!=0:# call the particular function for the user choice
46         if choice == 'b':
47             print(f"{binary(num)}")
48         elif choice == 'o':
49             print(f"{octal(num)}")
50         elif choice == 'h':
51             print(f"{hexadecimal(num)}")
52         else:
53             print("Invalid choice")
54     else: # if number is 0, there is no point for evaluation
55         print("0")
56
57 except ValueError:
58     logging.error("Invalid input", exc_info=True)
```

Ln 10, Col 11 Spaces: 4 UTF-8 LF () Python 3.10.12 64-bit

primes.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash

```
rishabh@TTNPL-rishabhsagar:~/Downloads$ python3 primes.py
Enter a number 5
b for binary, o for octal and h for hexadecimal
b
101
rishabh@TTNPL-rishabhsagar:~/Downloads$
```

Ln 10, Col 11 Spaces: 4 UTF-8 LF () Python 3.10.12 64-bit

primes.log
~/Downloads

Open ↕ Save ⋮ — 🔍 ✕

```
1 2025-02-03 17:01:43,450 - INFO - Converting 5 to binary
2 2025-02-03 17:01:43,450 - INFO - Binary result: 101|
```

Plain Text ▾ Tab Width: 8 ▾ Ln 2, Col 52 ▾ INS