

# ASSIGNMENT-1.4

2303A51546

BATCH-10

## TASK-1:

### PROMPT:

prime number check without using function

### CODE:

```
num = int(input("Enter a number: "))
```

```
if num > 1:
```

```
    for i in range(2, int(num/2)+1):
```

```
        if (num % i) == 0:
```

```
            print(num, "is not a prime number")
```

```
            break
```

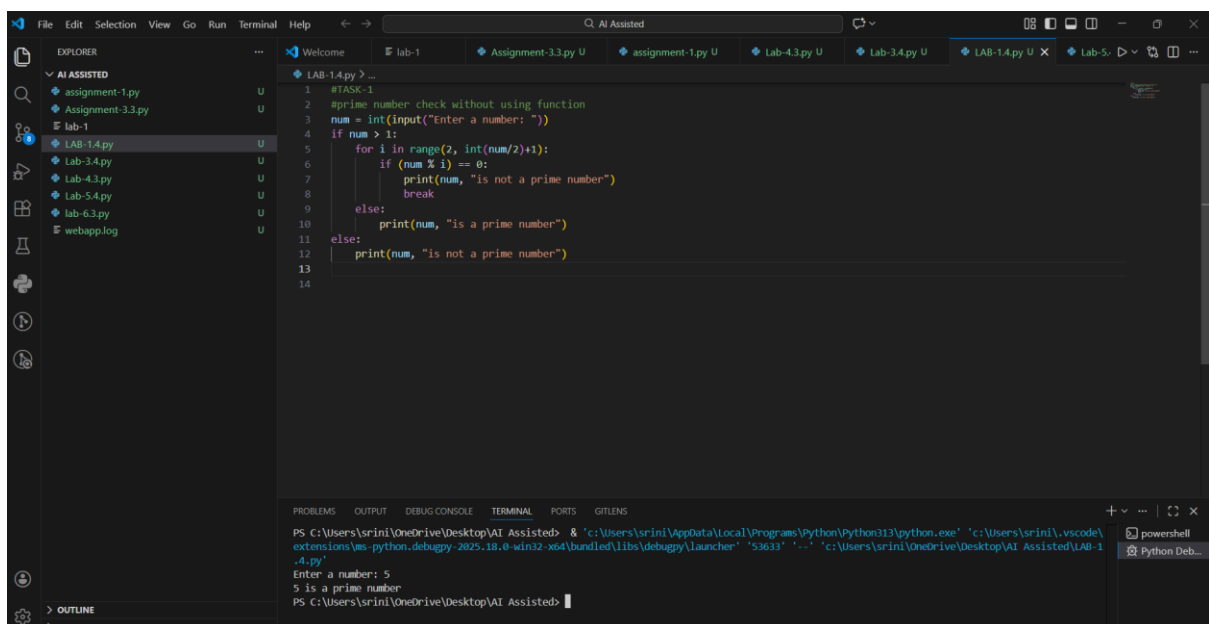
```
else:
```

```
    print(num, "is a prime number")
```

```
else:
```

```
    print(num, "is not a prime number")
```

### OUTPUT:



The screenshot shows a Visual Studio Code editor with a Python file named 'LAB-1.4.py'. The code implements a prime number check without using a function. The script prompts the user to enter a number, and in this case, the user entered '5'. The output shows that 5 is a prime number. The terminal at the bottom displays the command prompt and the execution of the script.

```
1 #TASK-1
2 #prime number check without using function
3 num = int(input("Enter a number: "))
4 if num > 1:
5     for i in range(2, int(num/2)+1):
6         if (num % i) == 0:
7             print(num, "is not a prime number")
8             break
9         else:
10            print(num, "is a prime number")
11 else:
12     print(num, "is not a prime number")
13
14
```

Terminal Output:

```
PS C:\Users\sriini\OneDrive\Desktop\AI Assisted> & 'c:\Users\sriini\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\sriini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '53633' '...' 'c:\Users\sriini\OneDrive\Desktop\AI Assisted\LAB-1.4.py'
Enter a number: 5
5 is a prime number
PS C:\Users\sriini\OneDrive\Desktop\AI Assisted>
```

## **Analysis:**

The user enters a number.

The program checks if the number is greater than 1 (since prime numbers are > 1).

It runs a loop from 2 to num/2 to see if any number divides it exactly.

If a divisor is found, it prints not prime and stops the loop using break.

If the loop finishes without finding any divisor, the for-else prints prime.

If the number is  $\leq 1$ , it directly prints not prime.

## **TASK-2:**

### **PROMPT:**

optimized the above code using function.

### **CODE:**

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if (num % i) == 0:
            return False
    return True
number = int(input("Enter a number: "))
if is_prime(number):
    print(number, "is a prime number")
else:
    print(number, "is not a prime number")
```

### **OUTPUT:**

```
13
14
15 #TASK-2
16 #optimized the above code using function.
17 def is_prime(num):
18     if num <= 1:
19         return False
20     for i in range(2, int(num**0.5) + 1):
21         if (num % i) == 0:
22             return False
23     return True
24
25 number = int(input("Enter a number: "))
26 if is_prime(number):
27     print(number, "is a prime number")
28 else:
29     print(number, "is not a prime number")
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GIT LENS

```
PS C:\Users\sriini\OneDrive\Desktop\AI Assisted> & 'c:\Users\sriini\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\sriini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '52847' '--' 'c:\Users\sriini\OneDrive\Desktop\AI Assisted\LAB-1-4.py'
Enter a number: 10
10 is not a prime number
PS C:\Users\sriini\OneDrive\Desktop\AI Assisted>
```

## Analysis:

A function `is_prime(num)` is defined.

If  $\text{number} \leq 1 \rightarrow$  returns False.

Loop runs from 2 to  $\sqrt{\text{num}}$ .

If any divisor is found  $\rightarrow$  returns False.

If no divisor is found  $\rightarrow$  returns True.

Main program takes input, calls the function, and prints result.

## TASK-3:

### PROMPT:

Fibonacci series without using function.

### CODE:

```
n = int(input("Enter the number of terms in Fibonacci series: "))
a, b = 0, 1
count = 0
if n <= 0:
    print("Please enter a positive integer.")
elif n == 1:
```

```

print("Fibonacci series up to", n, ":")
print(a)
else:
    print("Fibonacci series up to", n, ":")
    while count < n:
        print(a, end=' ')
        a, b = b, a + b
        count += 1

```

## OUTPUT:

```

28
29 #TASK-3
30 # Fibonacci series without using function.
31 n = int(input("Enter the number of terms in Fibonacci series: "))
32 a, b = 0, 1
33 count = 0
34 if n <= 0:
35     print("Please enter a positive integer.")
36 elif n == 1:
37     print("Fibonacci series up to", n, ":")
38     print(a)
39 else:
40     print("Fibonacci series up to", n, ":")
41     while count < n:
42         print(a, end=' ')
43         a, b = b, a + b
44         count += 1
45
46
47
48

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\sriini\OneDrive\Desktop\AI Assisted> & "c:\Users\sriini\AppData\Local\Programs\Python\Python313\python.exe" "c:\Users\sriini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher" "56537" "--" "c:\Users\sriini\OneDrive\Desktop\AI Assisted\LAB-1-4.py"

Enter the number of terms in Fibonacci series: 10

Fibonacci series up to 10 :

0 1 1 2 3 5 8 13 21 34

PS C:\Users\sriini\OneDrive\Desktop\AI Assisted>

## Analysis:

User enters number of terms n.

If  $n \leq 0 \rightarrow$  error message.

If  $n == 1 \rightarrow$  prints only 0.

Otherwise, a while loop runs n times:

- Prints current number a
- Updates values using  $a, b = b, a + b$

## **TASK-4:**

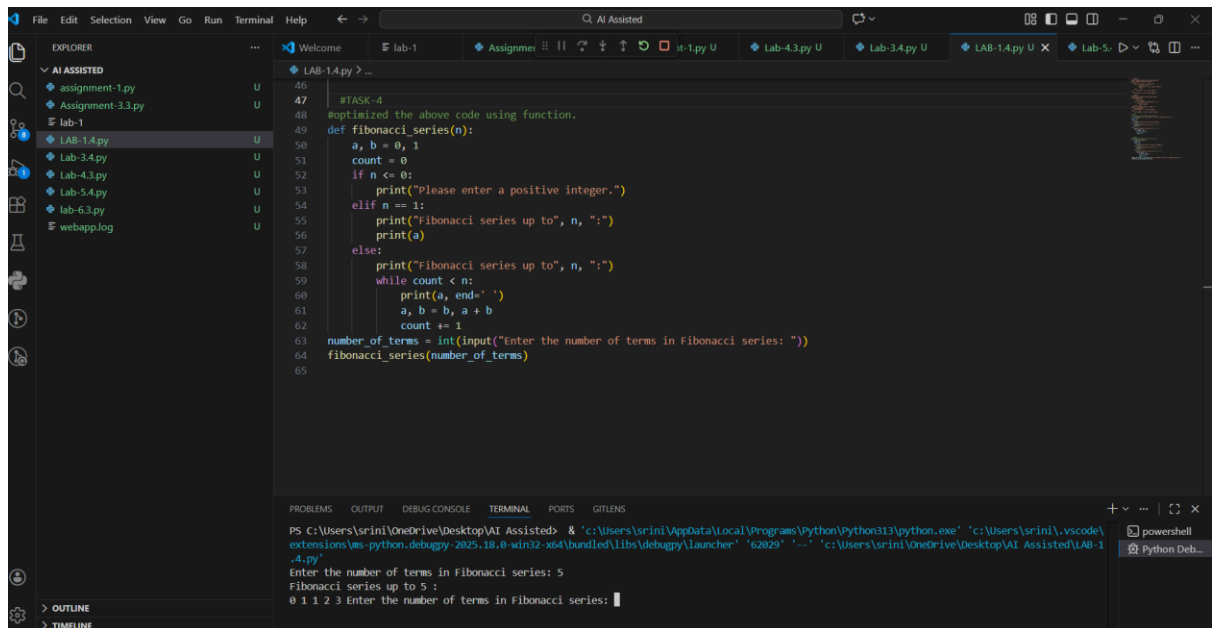
### **PROMPT:**

optimized the above code using function

### **CODE:**

```
def fibonacci_series(n):  
    a, b = 0, 1  
    count = 0  
    if n <= 0:  
        print("Please enter a positive integer.")  
    elif n == 1:  
        print("Fibonacci series up to", n, ":")  
        print(a)  
    else:  
        print("Fibonacci series up to", n, ":")  
        while count < n:  
            print(a, end=' ')  
            a, b = b, a + b  
            count += 1  
number_of_terms = int(input("Enter the number of terms in Fibonacci  
series: "))  
fibonacci_series(number_of_terms)
```

### **OUTPUT:**



```
46
47 #TASK-4
48 #optimized the above code using function.
49 def fibonacci_series(n):
50     a, b = 0, 1
51     count = 0
52     if n <= 0:
53         print("Please enter a positive integer.")
54     elif n == 1:
55         print("Fibonacci series up to", n, ":")
56         print(a)
57     else:
58         print("Fibonacci series up to", n, ":")
59         while count < n:
60             print(a, end=' ')
61             a, b = b, a + b
62             count += 1
63 number_of_terms = int(input("Enter the number of terms in Fibonacci series: "))
64 fibonacci_series(number_of_terms)
65
```

```
PS C:\Users\sirini\OneDrive\Desktop\AI Assisted> & 'c:\Users\sirini\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\sirini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '62029' '--' 'c:\Users\sirini\OneDrive\Desktop\AI Assisted\LAB-1.4.py'
Enter the number of terms in Fibonacci series: 5
Fibonacci series up to 5 :
0 1 1 2 3 Enter the number of terms in Fibonacci series: |
```

## Analysis:

Function fibonacci\_series(n) is defined.

It checks conditions for invalid, single-term, or multiple-term cases.

Prints the series inside the function.

Main program takes input and calls the function.

## TASK-5:

### PROMPT:

#Write a function to find the longest common prefix string amongst an array of strings.

# #If there is no common prefix, return an empty string "".

### CODE:

```
def longest_common_prefix(strs):
```

```
    if not strs:
```

```
        return ""
```

```
    prefix = strs[0]
```

```
    for s in strs[1:]:
```

```
        while s[:len(prefix)] != prefix and prefix:
```

```
            prefix = prefix[:-1]
```

```

    return prefix
strings = ["flower", "flow", "flight"]
result = longest_common_prefix(strings)
print("Longest common prefix:", result)
strings = ["dog", "racecar", "car"]
result = longest_common_prefix(strings)
print("Longest common prefix:", result)

```

## OUTPUT:

```

61     a, b = b, a + b
62     count += 1
63     number_of_terms = int(input("Enter the number of terms in Fibonacci series: "))
64     fibonacci_series(number_of_terms)'''
65
66
67 #TASK-5
68 #Write a function to find the longest common prefix amongst an array of strings.
69 # If there is no common prefix, return an empty string "".
70 def longest_common_prefix(strs):
71     if not strs:
72         return ""
73
74     prefix = strs[0]
75     for s in strs[1:]:
76         while s[:len(prefix)] != prefix and prefix:
77             prefix = prefix[:-1]
78
79     return prefix
80 strings = ["flower", "flow", "flight"]
81 result = longest_common_prefix(strings)
82 print("Longest common prefix:", result)
83 strings = ["dog", "racecar", "car"]
84 result = longest_common_prefix(strings)
85 print("Longest common prefix:", result)
86

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GIT LENS

```

PS C:\Users\sriini\OneDrive\Desktop\AI Assisted> & 'c:\Users\sriini\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\sriini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '63715' '--' 'c:\Users\sriini\OneDrive\Desktop\AI Assisted\LAB-1\4.py'
Longest common prefix: fl

```

## Analysis:

If the list is empty → return empty string.

Assume the first word is the prefix.

Compare it with each word in the list.

While the current word does not start with the prefix:

- Remove the last character of prefix (prefix = prefix[:-1])

When done, return the remaining prefix.