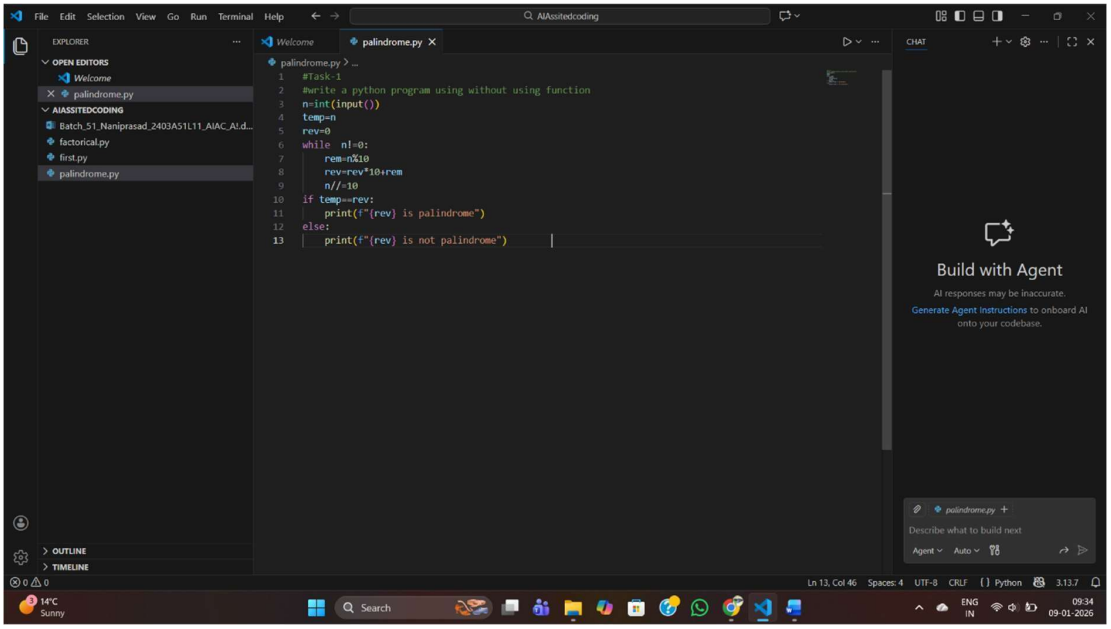
2203A51104

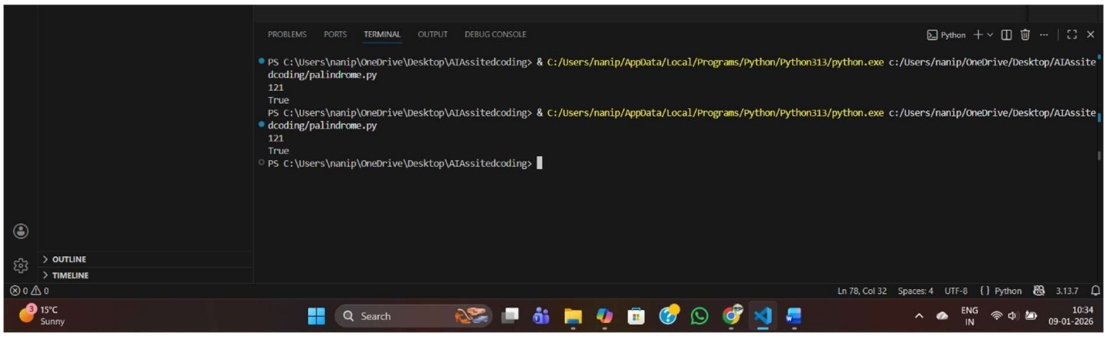
batch-52

#Task1

Write a python program for palindrome without using func on



Output:



Palindrome check steps for the given code

1. Read input:
    - Take an integer from the user and store it in n.
2. Store original number:
    - Copy n into temp so you can compare later a er reversing.

3. Ini alize reverse:
    - Set rev = 0. This will be built digit by digit into the reversed number.
4. Loop un l n becomes 0:
    - Keep extrac ng the last digit and removing it from n using integer division.
5. Extract last digit:
    - rem = n % 10 o This gives the rightmost digit of n.
6. Append digit to reversed number:
    - rev = rev * 10 + rem
    - Shi s exis ng digits in rev le and adds the new last digit.
7. Remove last digit from n:
    - n //= 10 o Drops the rightmost digit from n to process the next one.
8. End of loop:
    - When n becomes 0, rev now holds the full reversed number.

9.Compare original with reversed:

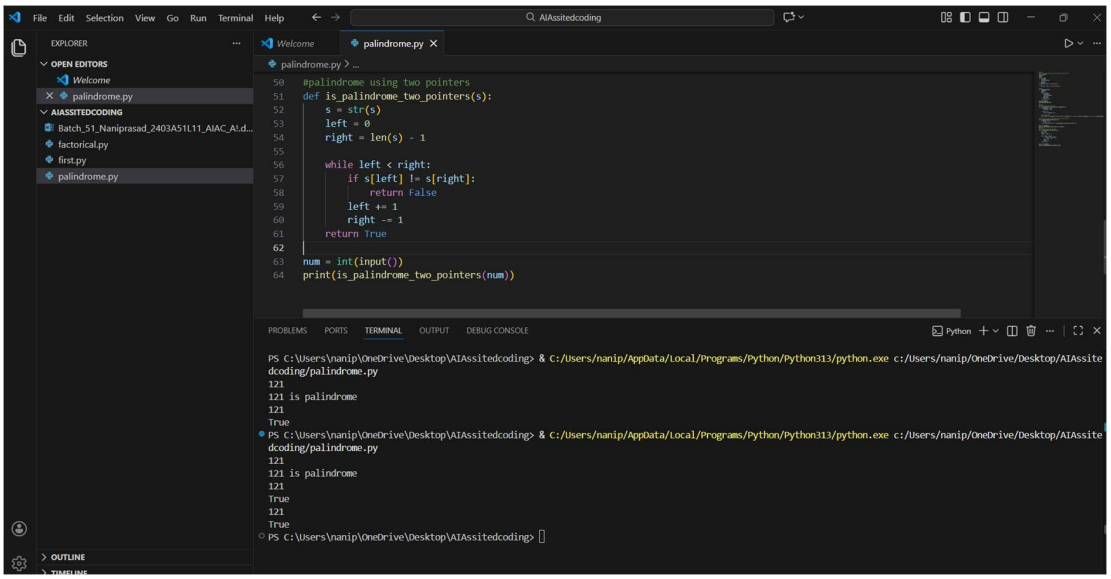
- If temp == rev, the original number reads the same backward → it's a palindrome. o Otherwise, it's not a palindrome.

10.Output result:

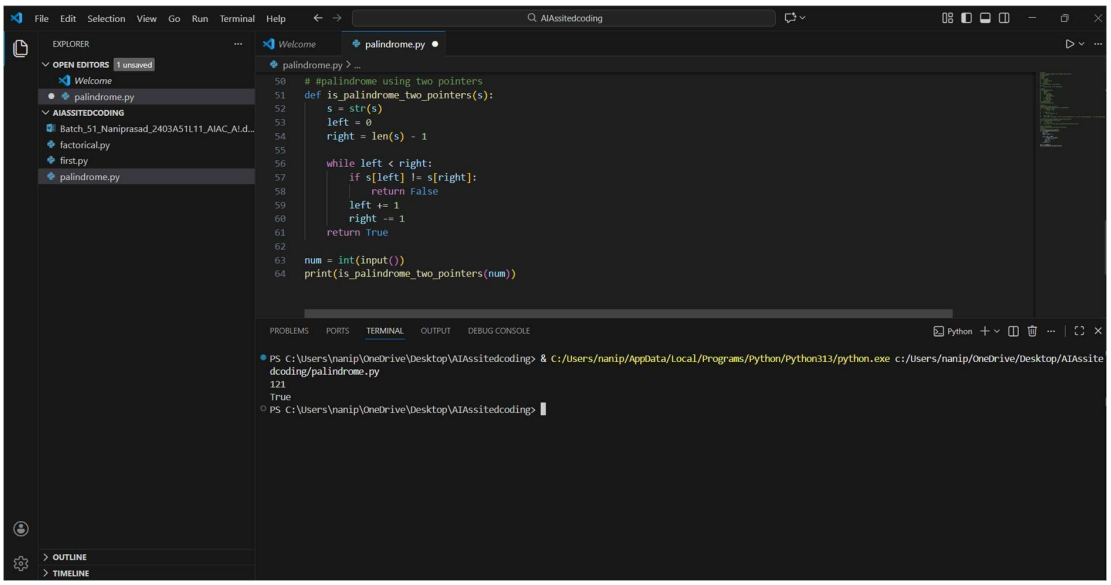
- Print "rev is palindrome" if equal, else "rev is not palindrome".

#Task2:

Write op mal solu on for palindrome solu on



Output:



Explaina on:

Create func on

Pass the input with some value

In two pointer if last and first value are equal then

Last-=1

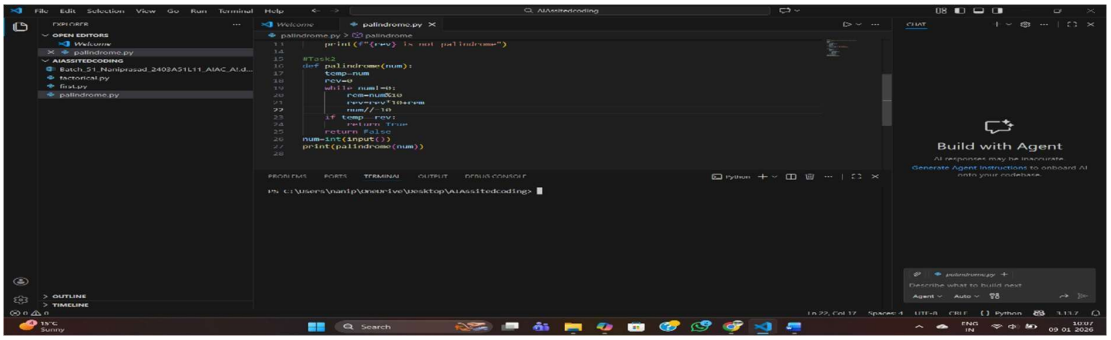
And first+=1

So if all index values are equal checking the last and first return True

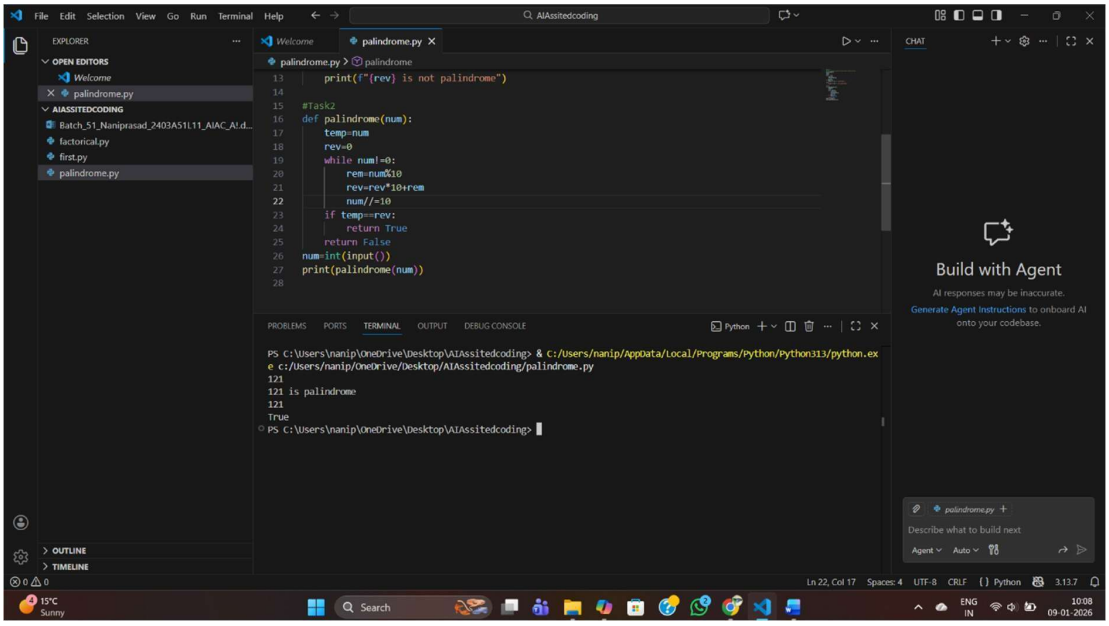
If not return False

#Task 3

Write python program for palindrome using func on



Output:



Explaina on:

Step-by-Step Explana on

1. Func on Defini on o def palindrome(num):
    - o A func on named palindrome is created that takes one argument num.
2. Store Original Number

- temp = num o The original number is stored in temp so we can compare later.

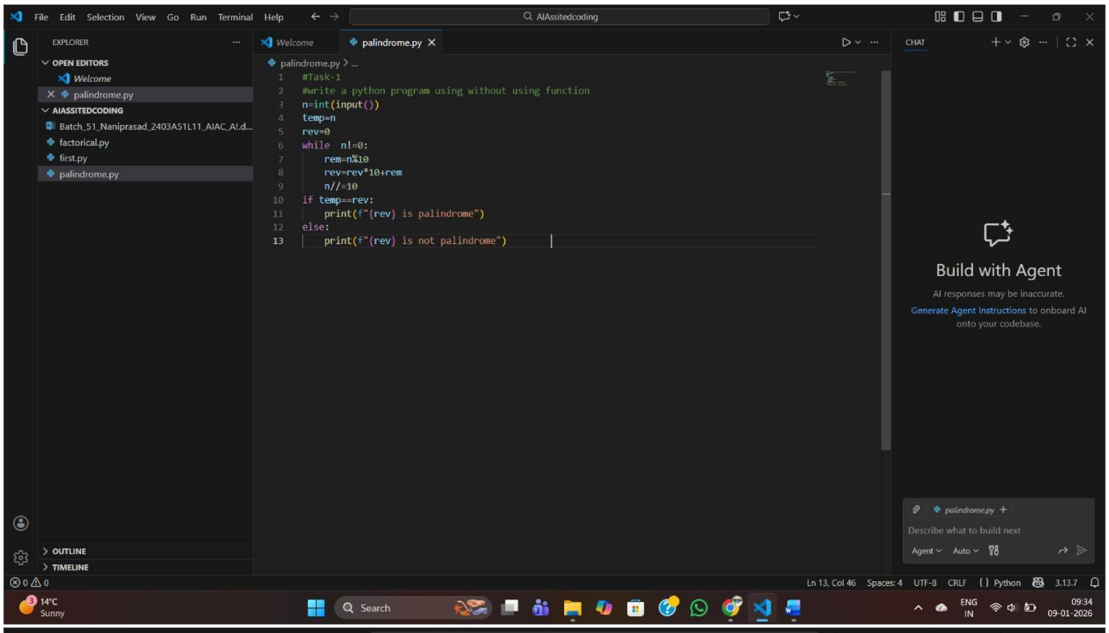
3. Ini alize Reverse
   - rev = 0 o This variable will hold the reversed number.
4. Loop to Reverse Number o while num != 0: → keep looping un l num becomes 0. o Inside the loop: o rem = num % 10 → extract the last digit.
   - rev = rev * 10 + rem → build the reversed number digit by digit.
   - num //= 10 → remove the last digit from num.
5. Check Palindrome o A er the loop ends, rev contains the reversed number. o Compare temp (original number) with rev.
   - If they are equal → return True.
   - Otherwise → return False.

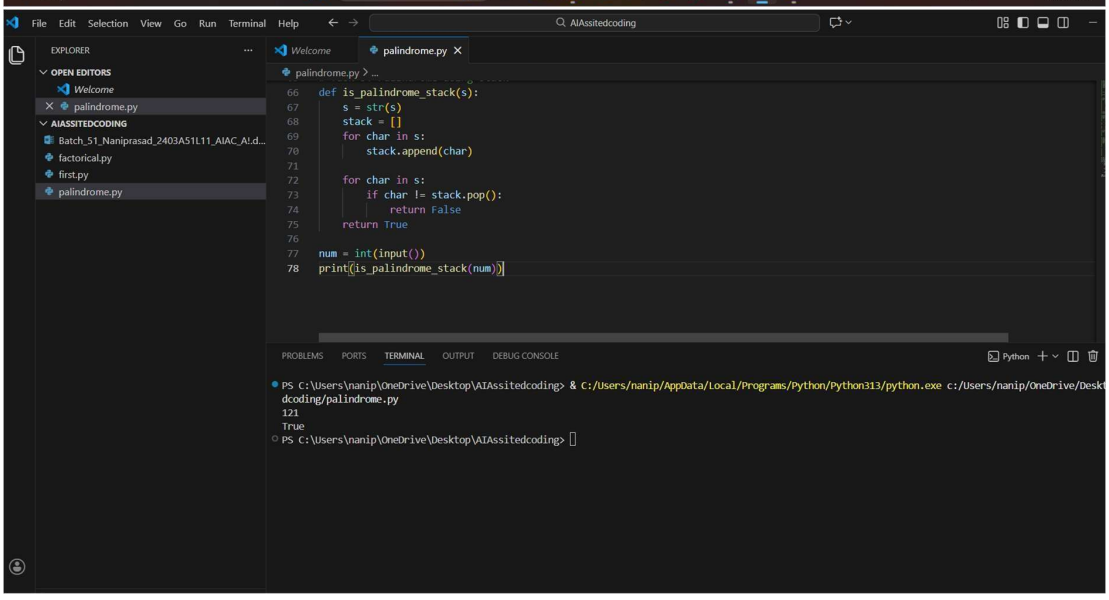
- Main Program
- num = int(input()) → take user input.
- print(palindrome(num)) → call the func on and print the result (True or False). Example Walkthrough Suppose input is 121:
- temp = 121, rev = 0
- Loop:
  - Itera on 1: rem = 1, rev = 1, num = 12 o Itera on 2: rem = 2, rev = 12, num = 1
  - Itera on 3: rem = 1, rev = 121, num = 0
- Loop ends → rev = 121
- Compare: temp == rev → 121 == 121 → True
- Output: True

If input is 123:

- Reverse becomes 321
- Compare: 123 != 321 → False
- Output: False #Task4:

Write Python program with using func on and without using func on





Output:

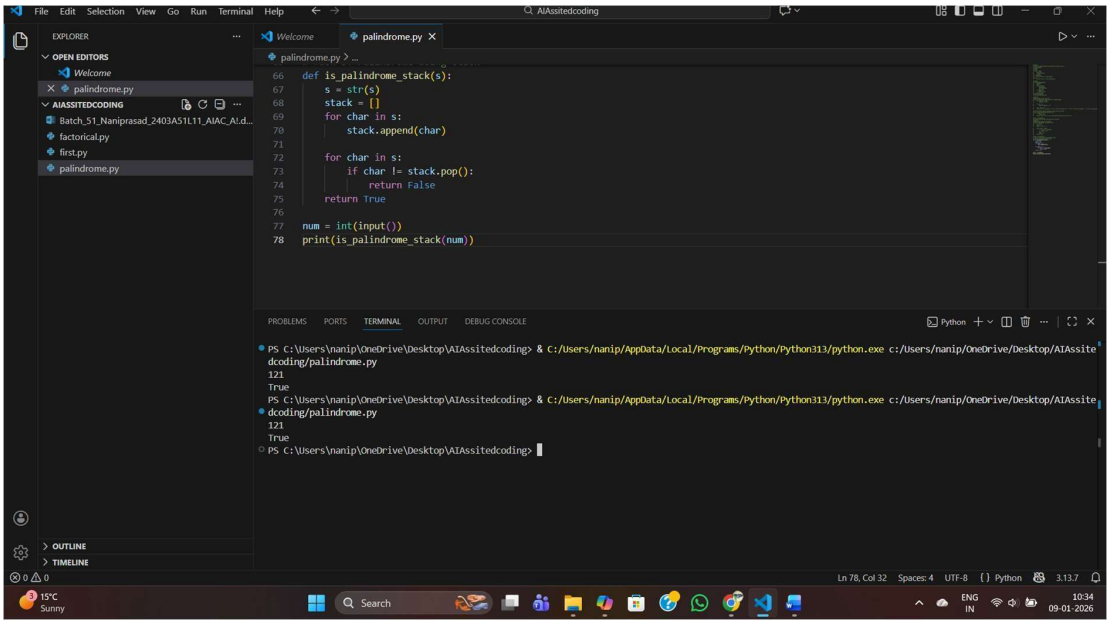
Step-by-Step

1. Input: User enters a number → stored in n.
2. Save original: temp = n keeps the original number safe.
3. Reverse logic:
    - Extract last digit using rem = n % 10.
    - Build reversed number: rev = rev * 10 + rem.
    - Remove last digit: n //= 10.
    - Repeat un l n becomes 0.
4. Compare: If temp == rev, the number is palindrome.
5. Output: Prints directly whether palindrome or not.

Step-by-Step

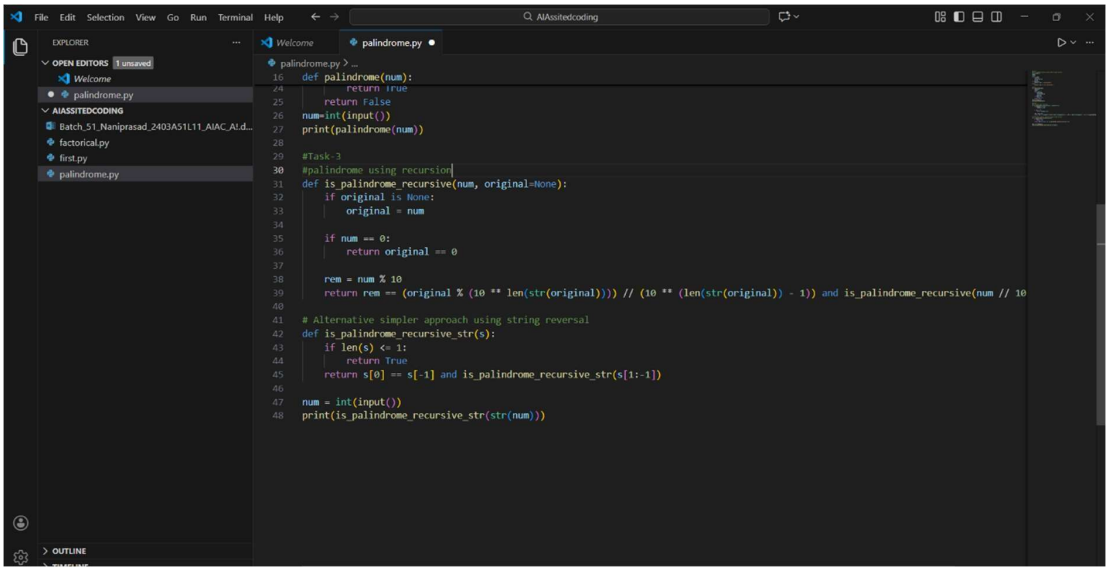
1. Func on defined: palindrome(num) encapsulates the logic.
2. Inside func on:
    - Store original number in temp. o Reverse the number using same loop logic. o Compare temp with rev.
    - Return True if palindrome, else False.
3. Main program:

- Take input from user.
- Call the func on: palindrome(num).
- Print the returned result (True or False).

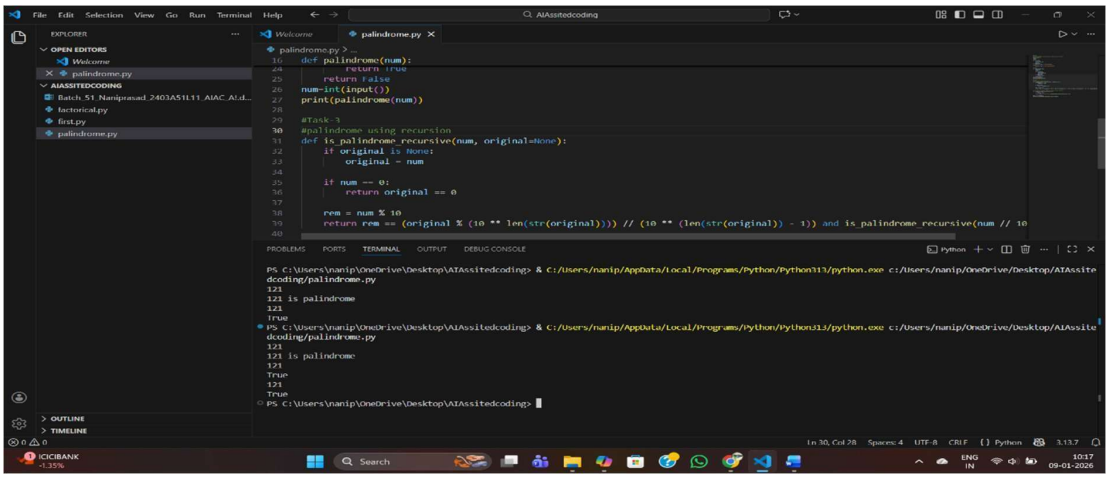


#Task5:

Write python program for palindrome using recursion



Output:



Step-by-Step Explana on

1. Convert number to string o str(num) turns the input number into a string. o Example: if user enters 121, then s = "121".
2. Recursive func on logic
   - o is_palindrome_recursive_str(s) checks if the string s is a palindrome.

3 Execu on Example: Input = 121

- o s = "121" o Step 1: Compare "1" (first) and "1" (last) → equal → recurse on "2".
- o Step 2: "2" has length 1 → base case → return True.
- o Final result: True.