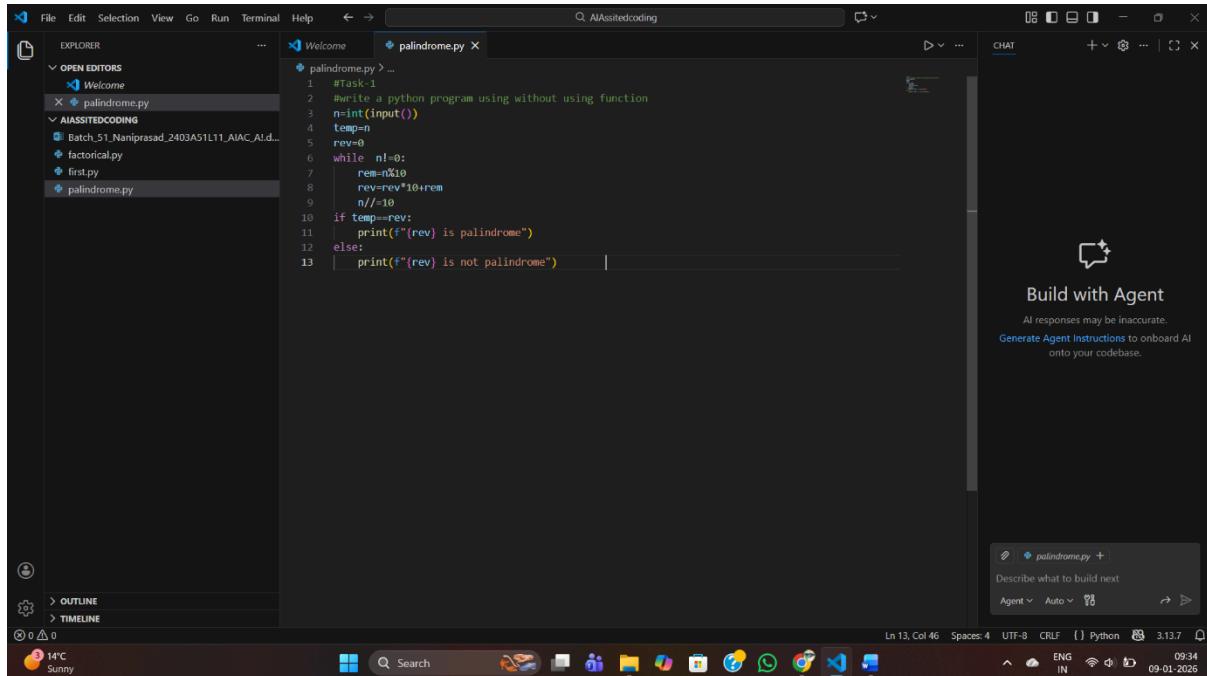


2403A51L29

batch-51

#Task1

Write a python program for palindrome without using function



```

File Edit Selection View Go Run Terminal Help < > Welcome palindrome.py
OPEN EDITORS
Welcome palindrome.py
AIASSITEDCODING
Batch_51_Naniprasad_2403A51L11_AIAC/AId...
factorial.py
first.py
palindrome.py

#Task-1
#write a python program using without using function
n=int(input())
temp=n
rev=0
while n!=0:
    rem=n%10
    rev=rev*10+rem
    n/=10
if temp==rev:
    print(f"{rev} is palindrome")
else:
    print(f"{rev} is not palindrome")

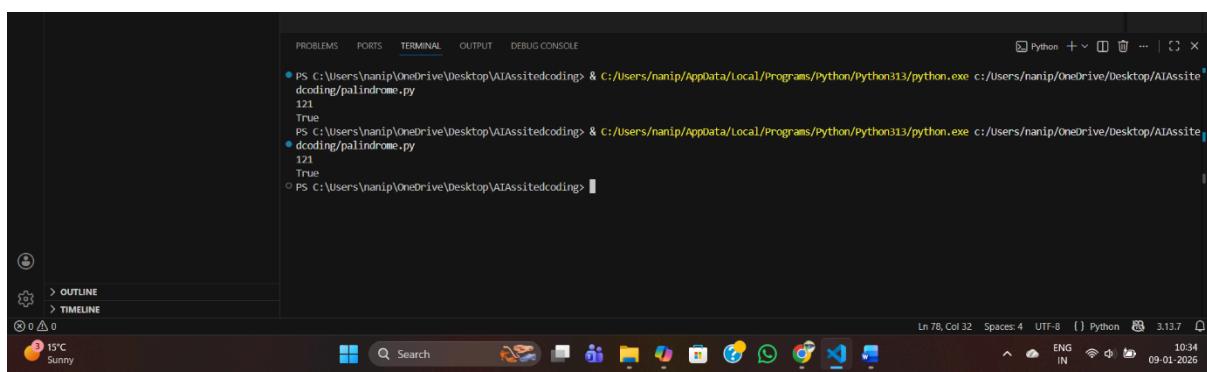
```

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

Describe what to build next
Agent Auto

Ln 13, Col 46 Spaces: 4 UTF-8 CRLF Python 3.13.7 09:34 14°C Sunny

Output:



```

PROBLEMS PORTS TERMINAL OUTPUT DEBUG CONSOLE Python + ... x
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & c:/users/nanip/appdata/local/programs/python/python313/python.exe c:/users/nanip/oneDrive/Desktop/AIAssistedcoding/palindrome.py
121
True
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & c:/users/nanip/appdata/local/programs/python/python313/python.exe c:/users/nanip/oneDrive/Desktop/AIAssistedcoding/palindrome.py
121
True
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding>

```

Ln 78, Col 32 Spaces: 4 UTF-8 ENG IN 10:34 15°C Sunny

Palindrome check steps for the given code

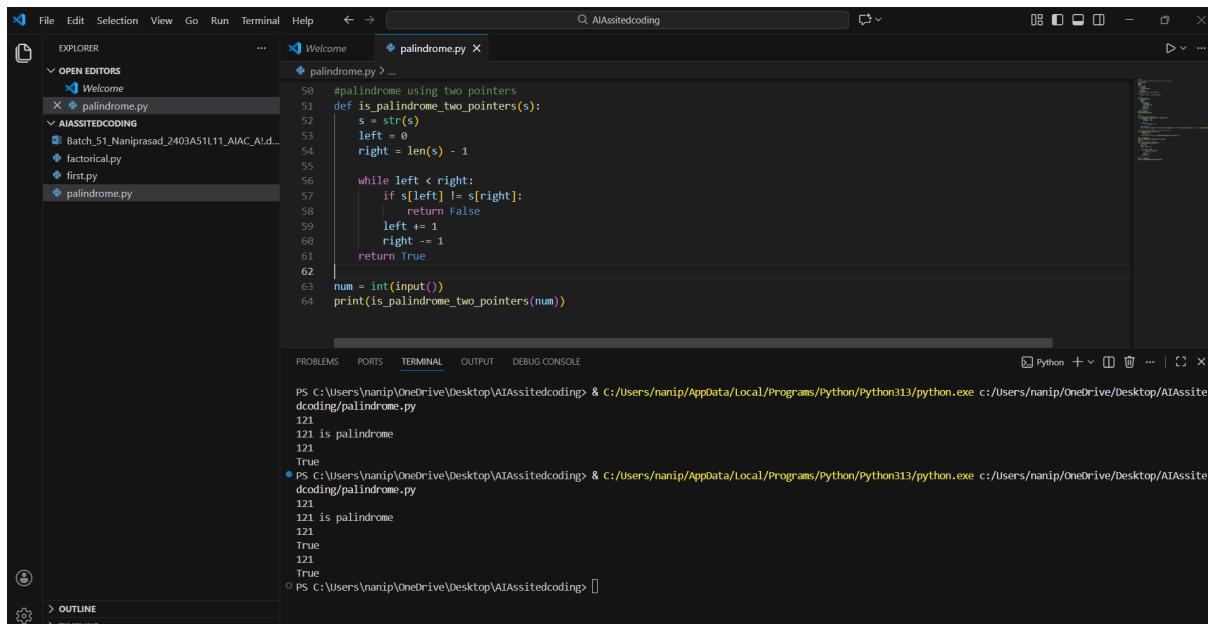
1. Read input:

- o Take an integer from the user and store it in n.

2. Store original number:
 - o Copy n into temp so you can compare later after reversing.
3. Initialize reverse:
 - o Set rev = 0. This will be built digit by digit into the reversed number.
4. Loop until n becomes 0:
 - o Keep extracting the last digit and removing it from n using integer division.
5. Extract last digit:
 - o `rem = n % 10`
 - o This gives the rightmost digit of n.
6. Append digit to reversed number:
 - o `rev = rev * 10 + rem`
 - o Shifts existing digits in rev left and adds the new last digit.
7. Remove last digit from n:
 - o `n /= 10`
 - o Drops the rightmost digit from n to process the next one.
8. **End of loop:**
 - o When n becomes 0, rev now holds the full reversed number.
9. **Compare original with reversed:**
 - o If `temp == rev`, the original number reads the same backward → it's a palindrome.
 - o Otherwise, it's not a palindrome.
10. **Output result:**
 - o Print “rev is palindrome” if equal, else “rev is not palindrome”.

#Task2:

Write optimal solution for palindrome solution



```

File Edit Selection View Go Run Terminal Help ← → Q AIAssistedcoding
EXPLORER OPEN EDITORS Welcome palindrome.py ...
AIASSISTEDCODING palindrome.py ...
ALIASSED CODING Batch_51_Naniprasad_2403A51L11_AIAC_Ald...
factorical.py first.py palindrome.py

50 #palindrome using two pointers
51 def is_palindrome_two_pointers(s):
52     s = str(s)
53     left = 0
54     right = len(s) - 1
55
56     while left < right:
57         if s[left] != s[right]:
58             return False
59         left += 1
60         right -= 1
61     return True
62
63 num = int(input())
64 print(is_palindrome_two_pointers(num))

```

PROBLEMS PORTS TERMINAL OUTPUT DEBUG CONSOLE

PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & c:/Users/nanip/AppData/Local/Programs/Python/Python313/python.exe c:/Users/nanip/OneDrive/Desktop/AIAssistedcoding/palindrome.py

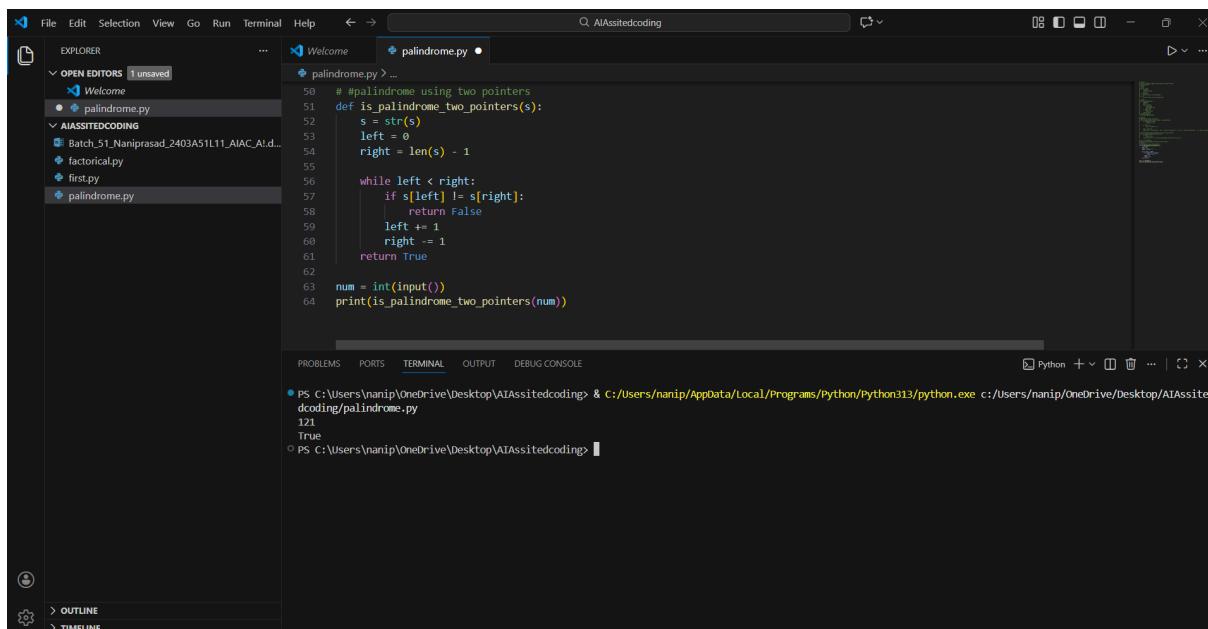
121
121 is palindrome
121
True

● PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & c:/Users/nanip/AppData/Local/Programs/Python/Python313/python.exe c:/Users/nanip/OneDrive/Desktop/AIAssistedcoding/palindrome.py

121
121 is palindrome
121
True
121
True

○ PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding>

Output:



```

File Edit Selection View Go Run Terminal Help ← → Q AIAssistedcoding
EXPLORER OPEN EDITORS palindrome.py ...
AIASSISTEDCODING palindrome.py ...
ALIASSED CODING Batch_51_Naniprasad_2403A51L11_AIAC_Ald...
factorical.py first.py palindrome.py

50 #palindrome using two pointers
51 def is_palindrome_two_pointers(s):
52     s = str(s)
53     left = 0
54     right = len(s) - 1
55
56     while left < right:
57         if s[left] != s[right]:
58             return False
59         left += 1
60         right -= 1
61     return True
62
63 num = int(input())
64 print(is_palindrome_two_pointers(num))

```

PROBLEMS PORTS TERMINAL OUTPUT DEBUG CONSOLE

● PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & c:/Users/nanip/AppData/Local/Programs/Python/Python313/python.exe c:/Users/nanip/OneDrive/Desktop/AIAssistedcoding/palindrome.py

121
121 is palindrome
121
True

○ PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding>

Explanation:

Create function

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 function

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a Help section for AI-Assisted Coding. The left sidebar displays project navigation with sections like GRADUATE, AI-ASSISTED, and ASSISTANT. The main code editor window contains a Python file named 'palindrome.py' with the following code:

```
1  print("Hello")
2  print("Welcome to AI-Assisted Coding")
3
4  num = int(input("Enter a number: "))
5
6  def isPalindrome(num):
7      temp = num
8      rev = 0
9      while num > 0:
10          rem = num % 10
11          rev = (rev * 10) + rem
12          num = num // 10
13
14      if temp == rev:
15          return True
16      else:
17          return False
18
19  print(isPalindrome(num))
```

The bottom status bar shows the path as PS C:\Users\harpal\Desktop\AI-AssistedCoding>, and the bottom right corner shows the date and time as 09.01.2020.

Output:

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar has sections for 'EXPLORER', 'OPEN EDITORS' (containing 'Welcome', 'palindrome.py', and 'first.py'), and 'AIASSISTEDCODING' (containing 'Batch_51_Naniprasad_2403A51L11_AIA_C_A1.d...'). The main area shows an open file 'palindrome.py' with the following code:

```
13     print(f"{rev} is not palindrome")
14
15 #Task2
16 def palindrome(num):
17     temp=num
18     rev=0
19     while num!=0:
20         rem=num%10
21         rev=rev*10+rem
22         num/=10
23     if temp==rev:
24         return True
25     return False
26 num=int(input())
27 print(palindrome(num))
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & c:/users/nanip/appdata/local/programs/python/python313/python.exe c:/users/nanip/OneDrive/Desktop/AIAssistedcoding/palindrome.py
121
121 is palindrome
121
True
○ PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding>
```

A floating panel on the right says 'Build with Agent' with the note 'AI responses may be inaccurate.' It includes a 'Generate Agent Instructions' button and a 'Describe what to build next' input field.

Explanation:

Step-by-Step Explanation

1. Function Definition

- `def palindrome(num):`
- A function named `palindrome` is created that takes one argument `num`.

2. Store Original Number

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

3. Initialize Reverse

- `rev = 0`
- This variable will hold the reversed number.

4. Loop to Reverse Number

- `while num != 0:` → keep looping until `num` becomes 0.
- Inside the loop:
- `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

- After the loop ends, `rev` contains the reversed number.
- 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 function and print the result (True or False).

Example Walkthrough

Suppose input is 121:

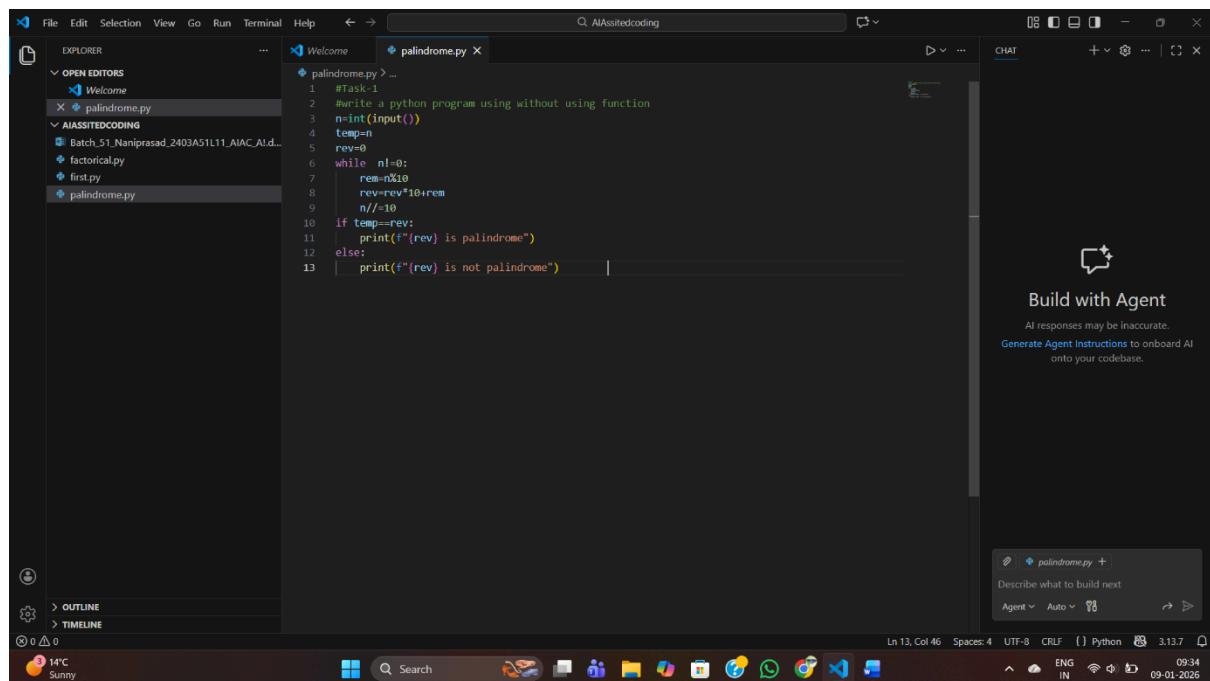
- `temp = 121, rev = 0`
- Loop:
 - Iteration 1: `rem = 1, rev = 1, num = 12`
 - Iteration 2: `rem = 2, rev = 12, num = 1`
 - Iteration 3: `rem = 1, rev = 121, num = 0`
- Loop ends → `rev = 121`
- Compare: `temp == rev` → `121 == 121` → True
- Output: True

If input is 123:

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

#Task4:

Write Python program with using function and without using function

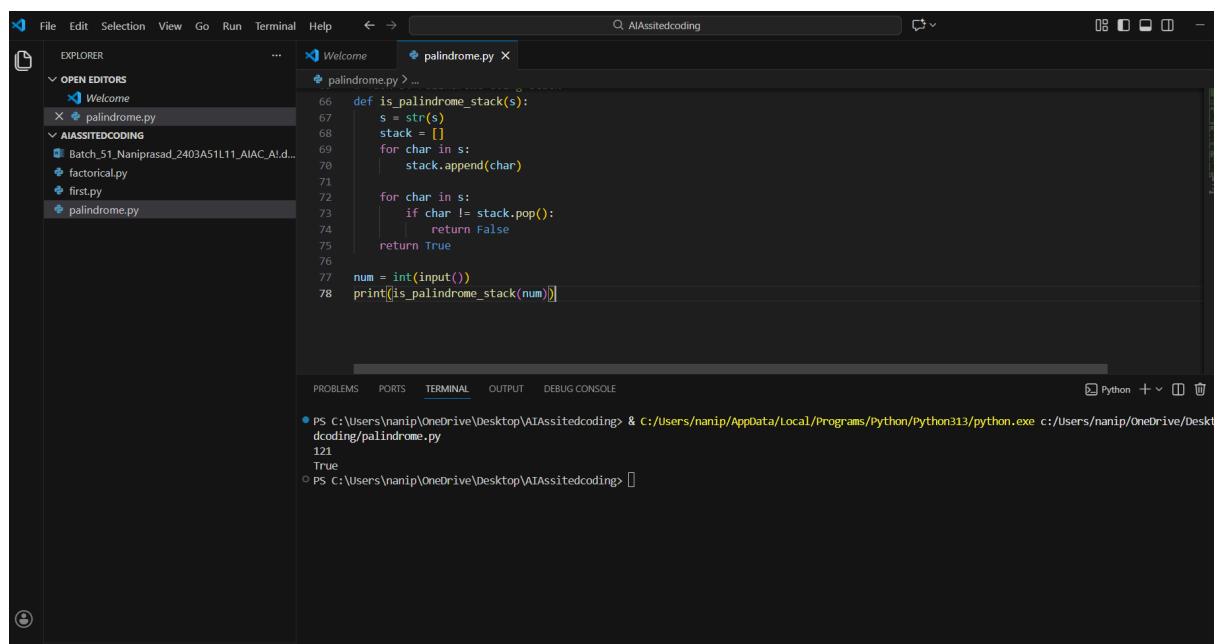


The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows multiple files including `Welcome`, `palindrome.py`, and several AI-assisted coding files.
- Editor:** Displays the code for `palindrome.py`:


```

1 #Task-1
2 #write a python program using without using function
3 n=int(input())
4 temp=n
5 rev=0
6 while n!=0:
7     rem=n%10
8     rev=rev*10+rem
9     n/=10
10 if temp==rev:
11     print(f"{rev} is palindrome")
12 else:
13     print(f"{rev} is not palindrome")
      
```
- AI Assistant Panel:** A sidebar titled "Build with Agent" with a message: "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase." It includes a text input field and a "Describe what to build next" button.
- Bottom Status Bar:** Shows file path (`palindrome.py`), line 13, column 46, spaces 4, encoding UTF-8, CRLF, Python 3.13.7, and date/time (09-01-2026).



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows multiple files including `Welcome`, `palindrome.py`, and several AI-assisted coding files.
- Editor:** Displays the code for `palindrome.py`:


```

66 def is_palindrome_stack(s):
67     s = str(s)
68     stack = []
69     for char in s:
70         stack.append(char)
71
72     for char in s:
73         if char != stack.pop():
74             return False
75     return True
76
77 num = int(input())
78 print(is_palindrome_stack(num))
      
```
- Terminal:** Shows the command line output:


```

PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & C:/Users/nanip/AppData/Local/Programs/Python/Python313/python.exe c:/Users/nanip/OneDrive/Desktop\AIAssistedcoding\palindrome.py
121
True
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding>
      
```

Output:

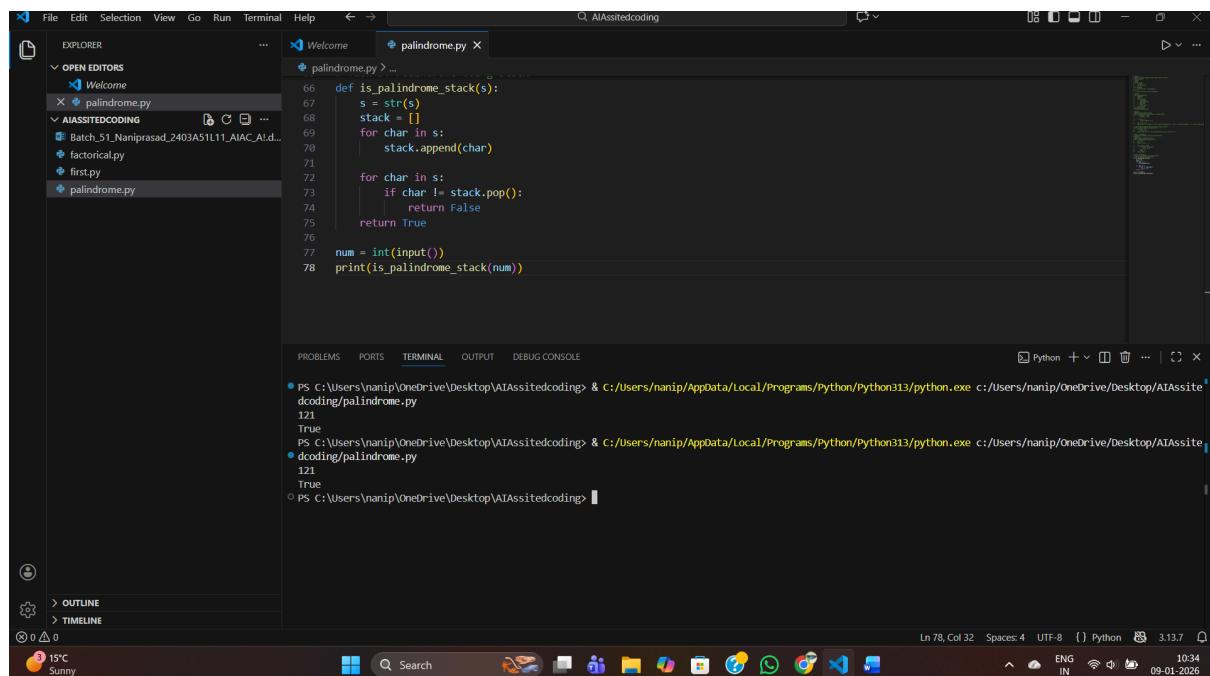
Step-by-Step

1. **Input:** User enters a number → stored in `n`.
2. **Save original:** `temp = n` keeps the original number safe.
3. **Reverse logic:**
 - o Extract last digit using `rem = n % 10`.

- Build reversed number: $\text{rev} = \text{rev} * 10 + \text{rem}$.
 - Remove last digit: $n \text{ } //= 10$.
 - Repeat until n becomes 0.
4. **Compare:** If $\text{temp} == \text{rev}$, the number is palindrome.
5. **Output:** Prints directly whether palindrome or not.

Step-by-Step

1. **Function defined:** `palindrome(num)` encapsulates the logic.
2. **Inside function:**
 - Store original number in `temp`.
 - Reverse the number using same loop logic.
 - Compare `temp` with `rev`.
 - Return True if palindrome, else False.
3. **Main program:**
 - Take input from user.
 - Call the function: `palindrome(num)`.
 - Print the returned result (True or False).



```

File Edit Selection View Go Run Terminal Help ← → Q AIAssistedCoding
EXPLORER OPEN EDITORS ... Welcome palindrome.py ...
OPEN EDITORS Welcome palindrome.py ...
AIASSISTEDCODING Batch_51_Naniprasad_2403A51L11_AIAC_Ald...
factorial.py first.py palindrome.py
Welcome palindrome.py ...
66 def is_palindrome_stack(s):
67     s = str(s)
68     stack = []
69     for char in s:
70         stack.append(char)
71
72     for char in s:
73         if char != stack.pop():
74             return False
75     return True
76
77 num = int(input())
78 print(is_palindrome_stack(num))

```

PROBLEMS PORTS TERMINAL OUTPUT DEBUG CONSOLE

```

PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & C:/Users/nanip/AppData/Local/Programs/Python/Python313/python.exe c:/users/nanip/OneDrive/Desktop/AIAssistedcoding/palindrome.py
121
True
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding> & C:/Users/nanip/AppData/Local/Programs/Python/Python313/python.exe c:/users/nanip/OneDrive/Desktop/AIAssistedcoding/palindrome.py
121
True
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedcoding>

```

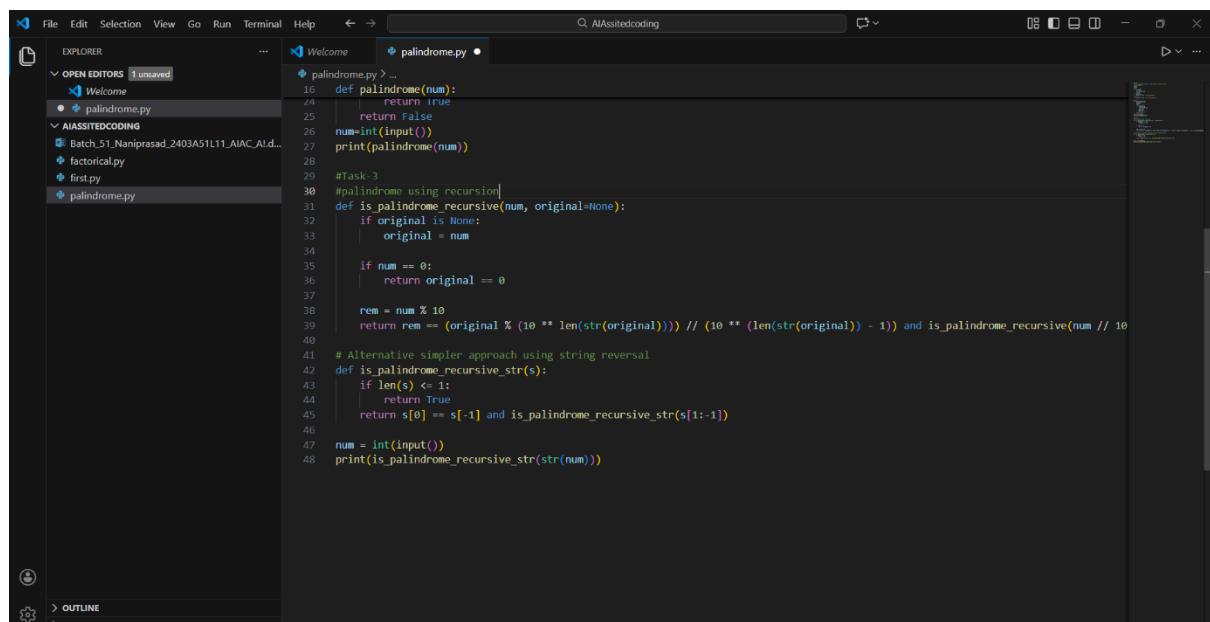
OUTLINE TIMELINE

15°C Sunny

Ln 78, Col 32 Spaces: 4 UTF-8 Python 3.13.7 10:34 09-01-2026

#Task5:

Write python program for palindrome using recursion



```

File Edit Selection View Go Run Terminal Help ← → Q AIAssistedCoding
EXPLORER OPEN EDITORS 1 unsaved ... Welcome palindrome.py ...
OPEN EDITORS Welcome palindrome.py ...
AIASSISTEDCODING Batch_51_Naniprasad_2403A51L11_AIAC_Ald...
factorial.py first.py palindrome.py
Welcome palindrome.py ...
16 def palindrome(num):
17     return True
18
19     return False
20 num=int(input())
21 print(palindrome(num))
22
23 #Task-3
24 #palindrome using recursion
25 def is_palindrome_recursive(num, original=None):
26     if original is None:
27         original = num
28
29     if num == 0:
30         return original == 0
31
32     rem = num % 10
33     return rem == (original % (10 ** len(str(original)))) // ((10 ** (len(str(original)) - 1)) and is_palindrome_recursive(num // 10
34
35     # Alternative simpler approach using string reversal
36     def is_palindrome_recursive_str(s):
37         if len(s) <= 1:
38             return True
39         return s[0] == s[-1] and is_palindrome_recursive_str(s[1:-1])
40
41     num = int(input())
42     print(is_palindrome_recursive_str(str(num)))

```

OUTLINE

Output:

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files like `Welcome`, `palindrome.py`, `factorial.py`, and `first.py`.
- Code Editor:** Displays the `palindrome.py` file with code for checking palindromes using recursion.
- Terminal:** Shows command-line output for running the script with different inputs.
- Bottom Bar:** Includes icons for file operations, search, and system status.

```
File Edit Selection View Go Run Terminal Help < > AIAssistedCoding

OPEN EDITORS
  Welcome
  palindrome.py
  factorial.py
  first.py
  palindrome.py

Welcome
16 def palindrome(num):
24     return True
25     return False
26 num=int(input())
27 print(palindrome(num))

#task 3
#palindrome using recursion
31 def isPalindromeRecursive(num, original=None):
32     if original is None:
33         original = num
34
35     if num == 0:
36         return original == 0
37
38     rem = num % 10
39     return rem == (original % (10 ** (len(str(original)) - 1))) // (10 ** (len(str(original)) - 1)) and isPalindromeRecursive(num // 10)

PROBLEMS PORTS TERMINAL OUTPUT DEBUG CONSOLE Python + □ ... ×

PS C:\Users\nanip\OneDrive\Desktop\AIAssistedCoding> & C:/Users/nanip/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nanip/OneDrive/Desktop/AIAssistedCoding/palindrome.py
121
121 is palindrome
121
True
True
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedCoding> & C:/Users/nanip/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nanip/OneDrive/Desktop/AIAssistedCoding/palindrome.py
121
121 is palindrome
121
True
True
True
PS C:\Users\nanip\OneDrive\Desktop\AIAssistedCoding>
```

Step-by-Step Explanation

1. Convert number to string

- `str(num)` turns the input number into a string.
 - Example: if user enters 121, then `s = "121"`.

2. Recursive function logic

- `is_palindrome_recursive_str(s)` checks if the string `s` is a palindrome.

3 Execution Example: Input = 121

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