**4.6** Given a string s, return the longest palindromic substring in S.

Example 1:

Input: s = "babad"

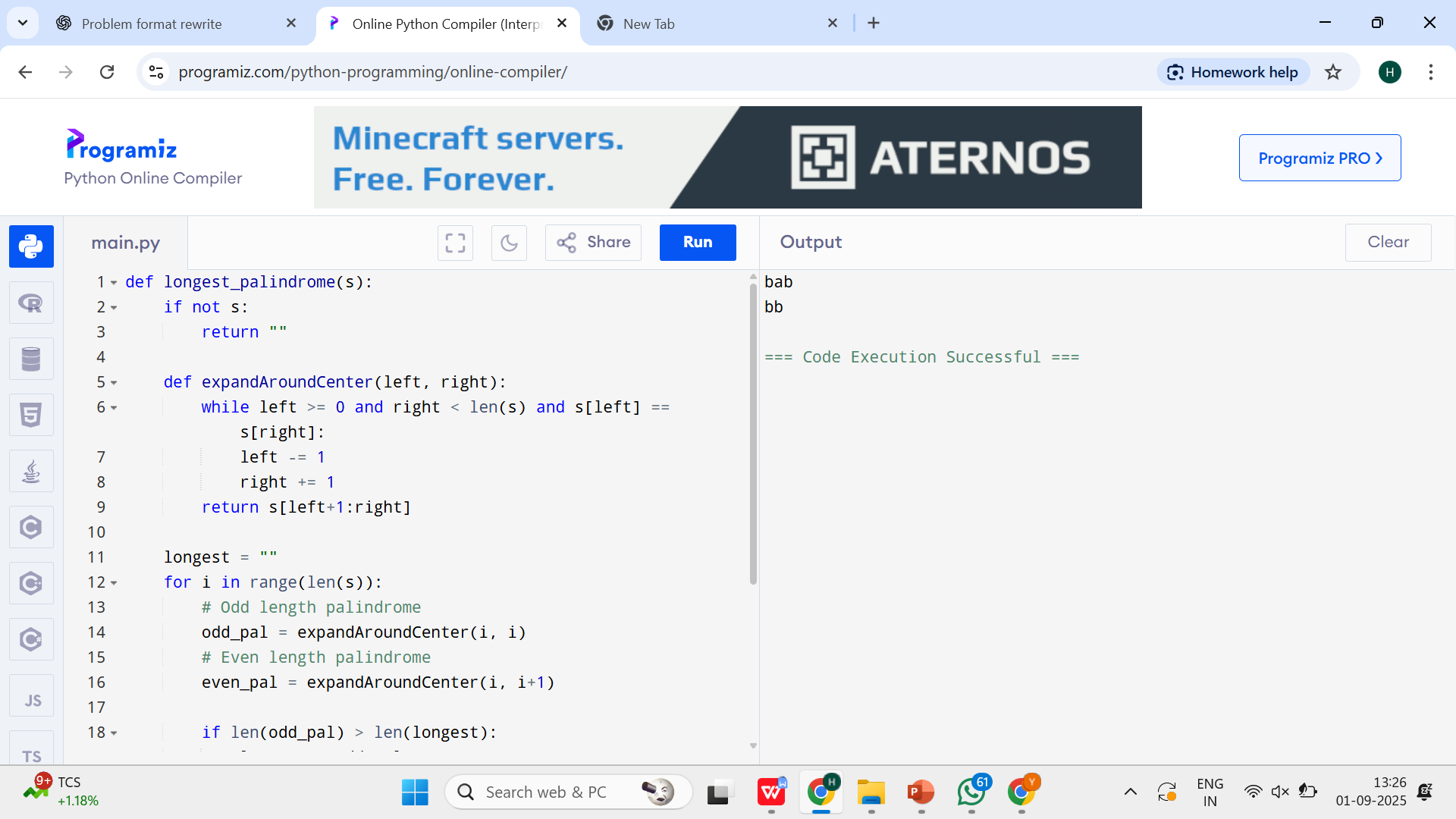
Output: "bab" Explanation: "aba" is also a valid answer.

**AIM**  
To write a program that finds the longest palindromic substring within a given string.

**ALGORITHM**

1. Input the string s.
2. Define a helper function expandAroundCenter(left, right) that:
3. Expands pointers left and right outward while the characters are equal and indices are within bounds.
4. Returns the substring found.
5. Iterate through each index i in the string:
6. Check for odd-length palindrome by expanding around (i, i).
7. Check for even-length palindrome by expanding around (i, i+1).
8. Update the longest palindrome if a longer one is found.
9. Return the longest palindrome.

**PROGRAM (Python) AND OUTPUT:-**



**INPUT**

s = "babad"

s = "cbbd"

**RESULT**  
The program successfully returns the longest palindromic substring for the given input strings.

**PERFORMANCE ANALYSIS**

**Time Complexity:** O(n²) — expanding from each center takes O(n), and there are O(n) centers.

**Space Complexity:** O(1) — no extra space apart from variables.