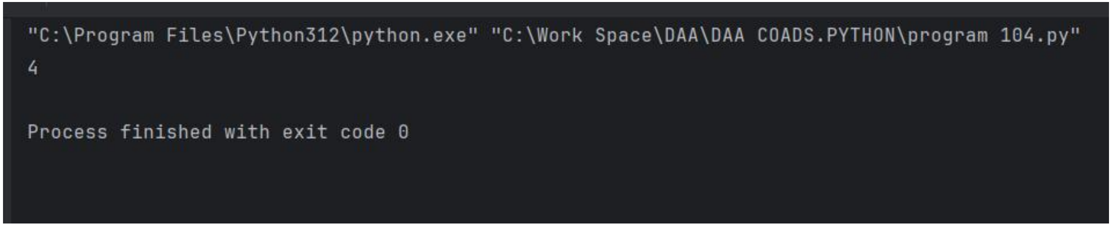
Program 104.Longest palindromic subsequence

Program:

```
def longest_palindromic_subsequence(s):
  n = len(s)
  # Create a 2D array to store the lengths of palindromic subsequences
  dp = [[0] * n for _ in range(n)]

  # Every single character is a palindrome of length 1
  for i in range(n):
    dp[i][i] = 1

  # Build the dp table
  # cl is the length of the substring
  for cl in range(2, n + 1):
    for i in range(n - cl + 1):
      j = i + cl - 1
      if s[i] == s[j] and cl == 2:
        dp[i][j] = 2
      elif s[i] == s[j]:
        dp[i][j] = dp[i + 1][j - 1] + 2
      else:
        dp[i][j] = max(dp[i][j - 1], dp[i + 1][j])

  return dp[0][n - 1]

# Example usage
s = "bbbab"
print(longest_palindromic_subsequence(s))  # Output: 4
```

Output:



```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA\DAA COADS.PYTHON\program 104.py"
4

Process finished with exit code 0
```

Time complexity:

O(n^2)