101. Longest palindromic subsequence

PROGRAM:

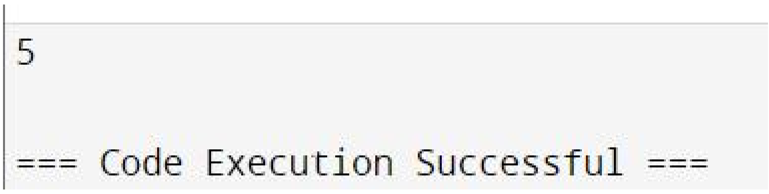
```
def longest_palindromic_subsequence(s):
  n = len(s)
  dp = [[0] * n for _ in range(n)]

  for i in range(n-1, -1, -1):
    dp[i][i] = 1
    for j in range(i+1, n):
      if s[i] == s[j]:
        dp[i][j] = 2 + dp[i+1][j-1]
      else:
        dp[i][j] = max(dp[i+1][j], dp[i][j-1])

  return dp[0][n-1]

# Example
s = "character"
print(longest_palindromic_subsequence(s))
```

OUTPUT:



```
5


=== Code Execution Successful ===
```

TIME COMPLEXITY:O(N^2)