102. Longest palindromic subsequence.

Program:

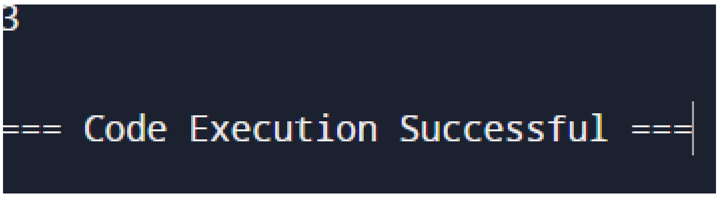
```
def longest_palindromic_subsequence(s):
    n = len(s)
    dp = [[0] * n for _ in range(n)]

    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 Usage
input_string = "babad"
result = longest_palindromic_subsequence(input_string)
print(result)
```

Output:



```
3

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

Time complexity: O(2n)