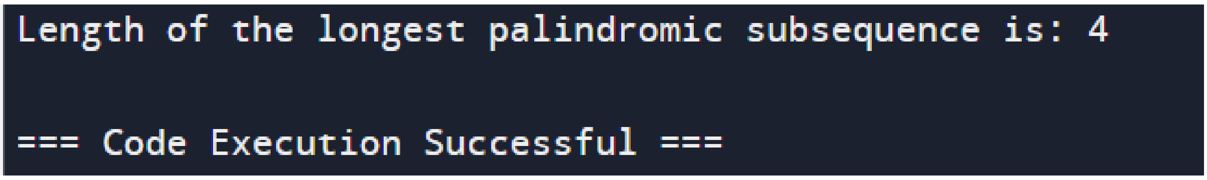
102.

PROGRAM:-

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

    # Subsequences of length 1 are palindromic of length 1
    for i in range(n):
        dp[i][i] = 1

    # Build the dp table
    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])

    # The length of the longest palindromic subsequence is in dp[0][n-1]
    return dp[0][n-1]

# Example usage
s = "bbbab"
length = longest_palindromic_subsequence(s)
print("Length of the longest palindromic subsequence is:", length)
```

OUTPUT:-



```
Length of the longest palindromic subsequence is: 4


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

TIME COMPLEXITY:-O(n^2)