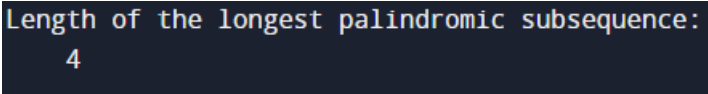


## 101. Longest Palindromic Subsequence

AIM: To find the length of ongest palindromic subsequence

PROGRAM:

```
def longest_palindromic_subsequence(s):  
    n = len(s)  
    dp = [[0] * n for _ in range(n)]  
    for i in range(n):  
        dp[i][i] = 1  
    for length in range(2, n + 1): # length of subsequence  
        for i in range(n - length + 1):  
            j = i + length - 1  
            if s[i] == s[j]:  
                dp[i][j] = dp[i+1][j-1] + 2  
            else:  
                dp[i][j] = max(dp[i+1][j], dp[i][j-1])  
  
    return dp[0][n-1]  
  
s = "bbbab"  
print("Length of the longest palindromic subsequence:", longest_palindromic_subsequence(s))
```

OUTPUT: 

TIME COMPLEXITY:  $O(n^2)$