Code:

```
/* if our start and end symbol match; we move forward start by 1 and move back end by 1
and add 2 to it ; b adad b ; here 1st and last matches therefore number of palindromic
subsequence in adad +2;
and if doesn't match we do check b adad or adad b; which ever has maximum we do consider
it as answer*/

class Solution{
  public:
    int longestPalinSubseq(string s)
    {
        int n=s.length();
      if(n==0 or n==1)
       return n;

      vector<vector<int>>dp(n,vector<int>(n,0));

      for(int gap=0;gap<n;gap++)
      {
         for(int i=0,j=gap;j<n && i<n-gap;i++,j++)
         {
            if(gap==0)
             dp[i][j]=1;
            else if(gap==1)
            {
               if(s[j]==s[i])
                dp[i][j]=2;
               else
                dp[i][j]=1;
            }
            else
            {
               if(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];
    }
};
```

Dry Run

Column denote end and row denote start

end

	b	b	a	b	c	b	c	a	b
b	1	2	2	3	3	5	5	5	7
b		1	1	3	3	3	3	5	7
a			1	1	1	3	3	5	5
b				1	1	3	3	3	5
c					1	1	3	3	3
b						1	1	1	3
c							1	1	1
a								1	1
b									1