2) a) We have an array like table that looks like $T(i)$ that is the maximum profit of $m_1, m_2, \dots, m_i$

b) $T(i) = p_i + \max\{T(j) + p(j)$ for $j < i$ and $m_i - m_j \geq k\}$
We have to set $T(1) = p(1)$

c) This table fills row-wise

d)
```
int T[n]
T[0] = p[0]
for (i=1; i<n; i++)
    T[i] = p[i]
    for (j=1; j<i; j++)
        if (m_i - m_j) ≥ K
            T[i] = max{T[i], p[i] + T[j]}
return max_{1≤i≤n} T[i]
```

e) This is a double for loop of size n so running time is $O(n^2)$

3) a) Take $T(i,k)$ to be the length of the longest possible palindrom. $s_i, s_{i+1}, \dots, s_j$

b) $T(i,j) = 0$ if $i > j$
$T(i,j) = 1$ if $i = j$
$T(i,j) = 2 + T(i+1, j-1)$ if $i < j$ and $s_i = s_j$
$T(i,j) = \max\{T(i+1, j), T(i, j-1)\}$ if $i < j$ and $s_i \neq s_j$

c) This Table starts filling the main diagonal and fills diagonally upto the top right cell.