```
temp = M[i,j]
for k = i+1 to j-1
    if ((M[i,k] + M[k,j])
        < temp)
        temp = M[i,k] + M[k,j]
        B[i,j] = k
M[i,j] = temp

return (M[0,n+1])
```

f. Give a recursive algorithm for reconstructing the solution, i.e., printing the cutting order.[2]

call print-recursive (B, 0, n+1) to print the sequence of cuts.

```
Print-recursive (B, i, j)
----------------------------
if ((i < j-1) and B[i,j] ≠ 0)
    k = B[i,j]
    print (k)
    print-recursive (B, i, k)
    print-recursive (B, k, j)
```

Q2. Recall the longest palindrome problem done in class. The dynamic programming solution is given here: You have to add the required code to reconstruct the optimal solution, i.e, to print out the longest palindrome. Provide a recursive function print_longest also [4]

```
Longest_palindrome(S, 1, n)
    Initialize array L[0..n, 0..n] to 0

    Initialize Array P[0..n, 0..n] to 0
    for i = 1 to n  P[i,i] = i

    for i=1 to n L[i,i]=1
    for offset=1 to n-1
        for i=1 to n-offset
            j=i+offset
            If ((S[i] == S[j]) and (L[i+1,j-1] == j-i-1))
                    L[I,j]=L[i+1,j-1] +2

                    P[i,j] = i

            else  L[I,j]=max(L[i+1,j], L[I,j-1])
                  if L[i+1, j] >= L[i, j-1]
                        P[i, j] = P[i+1, j]
                  else P[i,j] = P[i, j-1]

print_longest(P, 1,n)
```

```
print_longest(P,i,j)

if (i == j) print S[i]
else
  if i < j
      k = P[i, j]
      l = L[i, j]
      if (k > i)
         print_longest
            (P, k, k+l-1)
      else
         print S[k]
         print_longest
            (P, k+1, k+l-2)
         print S[k+l-1]
```

Alternatively the recursion could stop when k == i and then characters upto k+l-1 could be printed.

Q3. Consider ***the longest increasing subsequence problem***. Given a sequence S of n integers, you have to fing the longest increasing subsequence in it. For example, given S= 2, 5, 7, 3, 5, 6, 8, 11, 9 the longest increasing subsequence is 2, 3, 5, 6, 8, 9.

Give a formal problem statement and a state the optimal substructure in the problem. [2+2]

Given $S[1..n]$, a sequence of integers (elements $x \in \mathbb{Z}$), find the maximum positive integer K such that there exists a set of indices Z_1 to Z_K such that

① $\forall i : 1 \leq i \leq k$, $\exists y : S[y] = S[Z_k]$,

② $\forall_{i=1}^{k-1} \; 1 \leq Z_i < Z_{i+1} \leq S.\text{length}$.

③ $\forall_{i=1}^{k-1} \; S[Z_i] < S[Z_{i+1}]$

Consider element $S[0]$ to be MIN ($-\infty$ or minimum possible value) and $S[n+1]$ to be MAX ($+\infty$ or max possible value)

Define $L(i,j)$ to be the length of the maximum increasing subsequence in the sequence $S[i]$ to $S[j]$.

and let $I(i,j)$ to be the length of maximum increasing subsequence in the sequence $S[i]$ to $S[j]$, including both $S[i]$ and $S[j]$.

We need to find $I[0, n+1]$ which would be $L[1,n]+2$

Now, let S_K be a member of the longest increase