def max\_rev(price, n):  
 # build DP table  
 memo = []  
 for i in range(n + 1):  
 memo.append(0)  
  
 # fill table with best prices starting at each length  
 for i in range(1, n + 1):  
 memo[i] = -1  
 for j in range(i):  
 memo[i] = max(memo[i], price[j] + memo[i-j-1])  
  
 return memo[n]

PYTHON

def check\_sum(coins, sum):  
 memo = []  
  
 for i in range(sum + 1):  
 memo.append([])  
 for j in range(len(coins)):  
 memo[i].append(0)  
  
 for i in range(len(coins)):  
 memo[0][i] = 1  
  
 for i in range(1, sum+1):  
 for j in range(len(coins)):  
 x = memo[i - coins[j]][j] if i-coins[j] >= 0 else 0  
 y = memo[i][j-1] if j >= 1 else 0  
 memo[i][j] = x + y  
  
 return memo[sum][len(coins)-1] > 0

PYTHON

def check\_rev(text, start, end):  
 if start > end:  
 return ""  
 if memo[start][end] != -1:  
 return memo[start][end]  
  
 if text[start:end + 1] == text[start:end + 1][::-1]:  
 memo[start][end] = text[start:end + 1]  
 return text[start:end + 1]  
  
 # remove first character  
 sub1 = check\_rev(text, start + 1, end)  
  
 # remove last character  
 sub2 = check\_rev(text, start, end - 1)  
  
 memo[start][end] = sub1 if len(sub1) > len(sub2) else sub2  
 return memo[start][end]

PYTHON

def get\_row(i):  
 if memo[i] != -1:  
 return memo[i]  
  
 row = [1]  
 if i == 0:  
 return row  
  
 prev = get\_row(i - 1)  
 for j in range(len(prev) - 1):  
 row.append(prev[j] + prev[j + 1])  
  
 row.append(1)  
 memo[i] = row  
 return row

PYTHON