Problem – 6 : Matrix Chain Multiplication

```
def matrix_chain_multiplication(dimensions):
    n = len(dimensions)
    dp = [[0] * n for _ in range(n)]
    for i in range(n):
        dp[i][i] = 0
    for L in range(2, n):
        for i in range(1, n - L + 1):
            j = i + L - 1
            dp[i][j] = float('inf')
            for k in range(i, j):
                cost = dp[i][k] + dp[k+1][j] + dimensions[i-1] * dimensions[k] * dimensions[j]
                dp[i][j] = min(dp[i][j], cost)
    return dp[1][n-1]


dimensions = [10, 30, 5, 60]
result = matrix_chain_multiplication(dimensions)
print("Minimum number of multiplications:", result)
```