**Matrix multiplication using dynamic programming**

#include<bits/stdc++.h>

using namespace std;

int MatrixChainOrder(int p[], int n)

{

int m[n][n];

int i, j, k, L, q;

for (i = 1; i < n; i++)

m[i][i] = 0;

for (L = 2; L < n; L++)

{

for (i = 1; i < n - L + 1; i++)

{

j = i + L - 1;

m[i][j] = INT\_MAX;

for (k = i; k <= j - 1; k++)

{

q = m[i][k] + m[k + 1][j] +

p[i - 1] \* p[k] \* p[j];

if (q < m[i][j])

m[i][j] = q;

}

}

}

return m[1][n - 1];

}

int main()

{

int arr[] = {3, 4, 5, 1};

int size = sizeof(arr) / sizeof(arr[0]);

cout << "Minimum number of multiplications is "

<< MatrixChainOrder(arr, size);

getchar();

return 0;

}