 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering and Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject:</b> Design and Analysis of Algorithm	<b>Aim:</b> Implementing Matrix Chain Multiplication using Dynamic Programming Approach	
<b>Experiment No: 10</b>	<b>Date: 13/09/2025</b>	<b>Enrollment No: 92301733049</b>

## AIM: Implementing Matrix Chain Multiplication using Dynamic Programming Approach

```

#include <iostream>
#include <climits>
using namespace std;

// Function to find minimum number of multiplications
int matrixChainOrder(int p[], int n) {
    // dp[i][j] = minimum cost for multiplying matrices from i to j
    int dp[n][n];


    // Cost is 0 when multiplying one matrix
    for (int i = 1; i < n; i++)
        dp[i][i] = 0;

    // L is chain length
    for (int L = 2; L < n; L++) {
        for (int i = 1; i < n - L + 1; i++) {
            int j = i + L - 1;
            dp[i][j] = INT_MAX;

            for (int k = i; k < j; k++) {
                int cost = dp[i][k] + dp[k + 1][j] + p[i - 1] * p[k] * p[j];
                if (cost < dp[i][j])
                    dp[i][j] = cost;
            }
        }
    }

    return dp[1][n - 1]; // Minimum cost for full chain
}

```

 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering and Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject:</b> Design and Analysis of Algorithm	<b>Aim:</b> Implementing Matrix Chain Multiplication using Dynamic Programming Approach	
<b>Experiment No: 10</b>	<b>Date: 13/09/2025</b>	<b>Enrollment No: 92301733049</b>

```

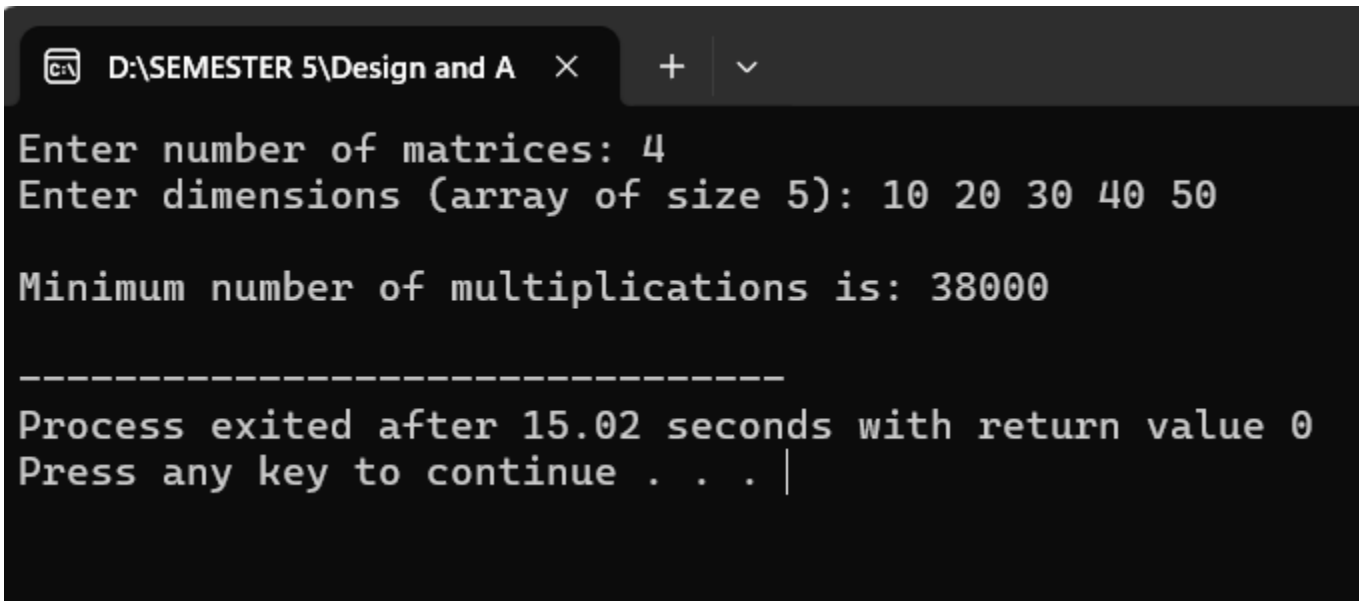
int main() {
    int n;
    cout << "Enter number of matrices: ";
    cin >> n;

    int p[n + 1];
    cout << "Enter dimensions (array of size " << n + 1 << "): ";
    for (int i = 0; i <= n; i++)
        cin >> p[i];

    cout << "\nMinimum number of multiplications is: "
        << matrixChainOrder(p, n + 1) << endl;

    return 0;
}

```



```

D:\SEMESTER 5\Design and A  ×  +  ∨
Enter number of matrices: 4
Enter dimensions (array of size 5): 10 20 30 40 50

Minimum number of multiplications is: 38000

-----
Process exited after 15.02 seconds with return value 0
Press any key to continue . . . |

```