Name: Shaan Agarwal
UID: 202130001
Branch: Computer Engineering
Batch: A
Subject: DAA
Experiment No: 4

Aim: Experiment Using Dynamic Programming Approach(Implementation Of Matrix Chain Multiplication)

Algorithm:

Matrix-Chain-Order(p)

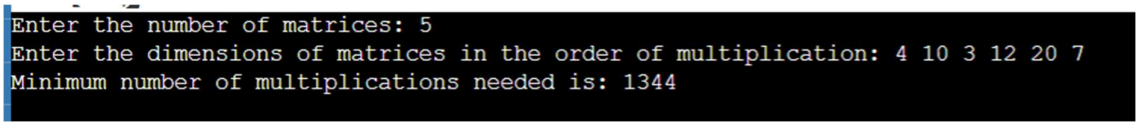
1. $n \leftarrow p.length - 1$
2. Let $m[1..n, 1..n]$ be a new matrix
3. for $i \leftarrow 1$ to $n$
4. $\quad m[i, i] \leftarrow 0$
5. for $L \leftarrow 2$ to $n$
6. $\quad$ for $i \leftarrow 1$ to $n-L+1$
7. $\quad\quad j \leftarrow i+L-1$
8. $\quad\quad m[i, j] \leftarrow \infty$
9. $\quad\quad$ for $k \leftarrow i$ to $j-1$
10. $\quad\quad\quad q \leftarrow m[i, k] + m[k+1, j] + p[i-1] \times p[k] \times p[j]$
11. $\quad\quad\quad$ if $q < m[i, j]$
12. $\quad\quad\quad\quad m[i, j] \leftarrow q$

13. return m

## Program:

```cpp
#include <bits/stdc++.h>
using namespace std;

int matrixChainOrder(int p[], int n) {
    int m[n][n];

    for (int i = 1; i < n; i++)
        m[i][i] = 0;

    for (int L = 2; L < n; L++) {
        for (int i = 1; i < n - L + 1; i++) {
            int j = i + L - 1;
            m[i][j] = INT_MAX;
            for (int k = i; k <= j - 1; k++) {
                int 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 n;
    cout << "Enter the number of matrices: ";
    cin >> n;
    int arr[n + 1];
    cout << "Enter the dimensions of matrices in the order of multiplication:
";
    for (int i = 0; i <= n; i++) {
        cin >> arr[i];
    }
    int size = sizeof(arr) / sizeof(arr[0]);
    cout << "Minimum number of multiplications needed is: " <<
matrixChainOrder(arr, size);
    return 0;
}
```

Output:




Conclusion: Thus, after performing the above experiment, I have understood the concept of Matrix Chain Multiplication, and have implemented the same using a C++ Program.