**MATRIX CHAIN MULTIPLICATION**

Let we have “n” number of matrices A1, A2, A3 ……… An and dimensions are d0 x d1, d1 x d2, d2 x d3 …………. dn-1 x dn.

Solving a chain of matrix that, Ai Ai+1 Ai+2 Ai+3 ……. Aj = (Ai Ai+1

Ai+2 Ai+3 ……. Ak ) (Ak+1 Ak+2 ……. Aj ) + di-1 dk dj where i <= k < j.

**CODING:**

#include<iostream>

#include<limits.h>

 using namespace std;

int MatrixChainMultiplication(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 n,i;

    cout<<"Enter number of matrices\n";

    cin>>n;

    n++;

   int arr[n];

    cout<<"Enter dimensions \n";

    for(i=0;i<n;i++)

    {

        cout<<"Enter position"<<i<<" :: ";

        cin>>arr[i];

    }

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

    cout<<"Minimum number of multiplications is "<<MatrixChainMultiplication(arr, size);

    return 0;

}

**OUTPUT:**

