**Program – 14**

**Aim – Write an algorithm and program to implement Matrix-Chain Multiplication.**

**Algorithm –**

1. n length[p]-1

2. for i ← 1 to n

3. do m [i, i] ← 0

4. for l ← 2 to n // l is the chain length

5. do for i ← 1 to n-l + 1

6. do j ← i+ l -1

7. m[i,j] ← ∞

8. for k ← i to j-1

9. do q ← m [i, k] + m [k + 1, j] + pi-1 pk pj

10. If q < m [i,j]

11. then m [i,j] ← q

12. s [i,j] ← k

13. return m and s.

**Source Code –**

#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 d"<<i<<" :: ";

        cin>>arr[i];

    }

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

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

    return 0;

}