Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India
(Autonomous College Affiliated to University of Mumbai)

Experiment No.	3
Aim	To implement matrix chain multiplication and also to compute its time complexity
Subject.	Design and Analysis of Algorithm
Name	Pratiksha Pravin Patil
UID No.	2022301016
Class & Division	Comp(A)

Theory:

Dynamic programming is a technique that breaks the problems into sub-problems, and saves the result for future purposes so that we do not need to compute the result again. The subproblems are optimized to optimize the overall solution is known as optimal substructure property. The main use of dynamic programming is to solve optimization problems. Here, optimization problems mean that when we are trying to find out the minimum or the maximum solution of a problem. The dynamic programming guarantees to find the optimal solution of a problem if the solution exists.

The definition of dynamic programming says that it is a technique for solving a complex problem by first breaking into a collection of simpler subproblems, solving each subproblem just once, and then storing their solutions to avoid repetitive computations.

Program:

```c
#include<stdio.h>
#include<limits.h>

// Matrix Ai has dimension p[i-1] x p[i] for i = 1..n

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;    //number of multiplications are 0(zero) when there is only one matrix

    //Here L is chain length. It varies from length 2 to length n.
    for (L=2; L<n; L++)
    {
        for (i=1; i<n-L+1; i++)
        {
            j = i+L-1;
            m[i][j] = INT_MAX;  //assigning to maximum value

            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;    //if number of multiplications found less that number will be updated.
                }
            }
        }
    }

    return m[1][n-1];   //returning the final answer which is M[1][n]
```

```c
                {
                    m[i][j] = q;    //if number of multiplications found less that number will be updated.
                }
            }
        }
    }

    return m[1][n-1];   //returning the final answer which is M[1][n]

}

int main()
{
    int n,i;
    printf("Enter number of matrices\n");
    scanf("%d",&n);

    n++;

    int arr[n];

    printf("Enter dimensions \n");

    for(i=0;i<n;i++)
    {
        printf("Enter d%d :: ",i);
        scanf("%d",&arr[i]);
    }

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

    printf("Minimum number of multiplications is %d ", MatrixChainMultiplication(arr, size));

    return 0;
}
```

Output:-

```
Enter number of matrices
4
Enter dimensions
Enter d0 :: 1
Enter d1 :: 2
Enter d2 :: 4
Enter d3 :: 5
Enter d4 :: 6
Minimum number of multiplications is 58

...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion: This I have studied and successfully implemented matrix chain multiplication.