

Name	Harsh Chandra
UID no.	2021700013
Experiment No.	5

AIM:	Implement matrix chain multiplication.
Program 1	
PROBLEM STATEMENT :	Matrix Chain Multiplication using dynamic programming.
ALGORITHM/ THEORY:	MATRIX-CHAIN-ORDER (p) 1. $n \leftarrow \text{length}[p]-1$ 2. for $i \leftarrow 1$ to n 3. do $m[i, i] \leftarrow 0$ 4. for $l \leftarrow 2$ to n // l is the chain length 5. do for $i \leftarrow 1$ to $n-l + 1$ 6. do $j \leftarrow i+ l -1$ 7. $m[i, j] \leftarrow \infty$ 8. for $k \leftarrow i$ to $j-1$ 9. do $q \leftarrow m[i, k] + m[k + 1, j] + p_{i-1} p_k p_j$ 10. If $q < m[i, j]$ 11. then $m[i, j] \leftarrow q$ 12. $s[i, j] \leftarrow k$ 13. return m and s

PROGRAM:

```
#include <stdio.h>
#include <limits.h>
void printParenthesis(int i, int j, int n, int *bracket, char *name)
{
    if (i == j)
    {
        printf("%c", (*name)++);
        return;
    }
    printf("(");
    printParenthesis(i, *((bracket + i * n) + j), n, bracket, name);
    printParenthesis(*((bracket + i * n) + j) + 1, j, n, bracket, name);
    printf(")");
}

void matrixChainOrder(int p[], int n)
{
    int m[n][n];
    int bracket[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;
                    bracket[i][j] = k;
                }
            }
        }
    }

    char name = 'A';
    printf("Optimal Parenthesization is : ");
```

```

        printParenthesis(1, n - 1, n, (int *)bracket, &name);
        printf("\nOptimal Cost is : %d\n",m[1][n - 1]);
    }
    int main()
    {
        int n;
        printf("Enter no. of matrices: ");

        scanf("%d", &n);
        int arr[n];
        printf("Enter the dimensions: ");
        for (int i = 0; i < n; i++)
            scanf("%d",&arr[i]);
        matrixChainOrder(arr, n);
        return 0;
    }

```

RESULT:

```

students@CE-Lab7-603-U10:~/Desktop$ gcc exp5MCM.c
students@CE-Lab7-603-U10:~/Desktop$ ./a.out
Enter no. of matrices: 5
Enter the dimensions: 10 20 30 40 50
Optimal Parenthesization is : ((AB)C)D)
Optimal Cost is : 38000
students@CE-Lab7-603-U10:~/Desktop$ █

```

CONCLUSION:

From this experiment I understood matrix chain multiplication using dynamic programming.