

Name	Siddhesh Sonar
UID no.	2021700063
Experiment No.	5

AIM:	To implement Matrix Chain Multiplication
Program	
PROBLEM STATEMENT :	To implement Matrix Chain Multiplication
ALGORITHM/ THEORY:	<p>Matrix chain multiplication is an optimization problem concerning the most efficient way to multiply a given sequence of matrices. The problem is not actually to perform the multiplications, but merely to decide the sequence of the matrix multiplications involved.</p> <pre> Begin define table minMul of size n x n, initially fill with all 0s for length := 2 to n, do for i:=1 to n-length, do j := i + length - 1 minMul[i, j] := ∞ for k := i to j-1, do q := minMul[i, k] + minMul[k+1, j] + array[i-1]*array[k]*array[j] if q < minMul[i, j], then minMul[i, j] := q done done done return minMul[1, n-1] End </pre>

PROGRAM:

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

int MatrixChainOrder(int arr[], int n)
{
    int mat[n][n], len, c, j;

    for (int i = 1; i < n; i++) {
        for (int j = 1; j < n; j++) {
            if(i >= j) {
                mat[i][j] = 0;
            }
        }
    }

    for (len = 2; len < n; len++) {
        for (int i = 1; i < n - len + 1; i++) {
            j = i + len - 1;
            mat[i][j] = INT_MAX;
            for (int k = i; k <= j - 1; k++) {
                c = mat[i][k] + mat[k + 1][j] + arr[i - 1] *
arr[k] * arr[j];
                if (c < mat[i][j]) {
                    mat[i][j] = c;
                }
            }
        }
    }

    for(int i=1;i<n;i++) {
        printf("\n");
        for(int j=1;j<n;j++) {
            printf("%d\t",mat[i][j]);
        }
    }

    return mat[1][n - 1];
}

int main()
{
    int n;
```

```

//int arr[] = { 1, 2, 3, 4 };
//int n = sizeof(arr) / sizeof(arr[0]);
printf("\nEnter the size of array: ");
scanf("%d", &n);
int arr[n];
for(int i = 0 ; i < n ; i++) {
    printf("\nEnter the %d element of the array: ", i+1);
    scanf("%d", &arr[i]);
}
printf("\n\nMinimum number of multiplications: %d ",
        MatrixChainOrder(arr, n));
return 0;
}

```

RESULT:

```

1#include <limits.h>
2#include <stdio.h>
3
4int MatrixChainOrder(int arr[], int n)
5{
6    int mat[n][n], len, c, j;
7
8    for (int i = 1; i < n; i++) {
9        for (int j = i + 1; j <= n; j++) {
10            if (i >= j) {
11                mat[i][j] = 0;
12            }
13        }
14    }
15
16    for (len = 2; len <= n; len++) {
17        for (int i = 1; i <= n - len + 1; i++) {
18            j = i + len - 1;
19            mat[i][j] = INT_MAX;
20            for (int k = i; k <= j - 1; k++) {
21                c = mat[i][k] + mat[k + 1][j] + arr[i - 1] *
22                arr[k] * arr[j];
23                if (c < mat[i][j]) {
24                    mat[i][j] = c;
25                }
26            }
27        }
28    }
29    for (int i = 1; i <= n; i++) {
30        printf("\n");
31        for (int j = 1; j <= n; j++) {
32            printf("%d\t", mat[i][j]);
33        }
34    }
35}

```

Terminal Output:

```

itdept@itdept-OptiPlex-3010:~$ gcc -o mat matrix_chain.c
itdept@itdept-OptiPlex-3010:~$ ./mat
Enter the size of array: 4
Enter the 1 element of the array: 1
Enter the 2 element of the array: 2
Enter the 3 element of the array: 3
Enter the 4 element of the array: 4
0      6      18
0      0      24
0      0      0
Minimum number of multiplications: 18 itdept@itdept-OptiPlex-3010:~$

```

CONCLUSION:

Successfully understood Matrix Chain Multiplication algorithm and implemented it in C program to find the minimum number of steps for multiplication of 2 matrices.