Assignment - 5:

WAP to multiply a chain of matrices optimally using iterative version of Dynamic Programming approach. Check your program for the sequence of 4 matrices <M1,M2,M3,M4> whose sequence of dimension is <10,20,50,1,100>.

Algorithm –

Procedure: printOptimalParent(int s[][n], int i, int j)
Inputs: s[][n] – array (or matrix) of size n.
i – value = 1.
j – value = n – 1.
Output: It will return the optimal parent.

```
1. if i == j do
2.         print ("M %d", i).
3. else
4.         print (" ( ").
5.         call printOptimalParent(s, i, s[i][j]).
6.         call printOptimalParent(s, s[i][j] + 1, j).
7.         print (" ) ").
8. end if.
```

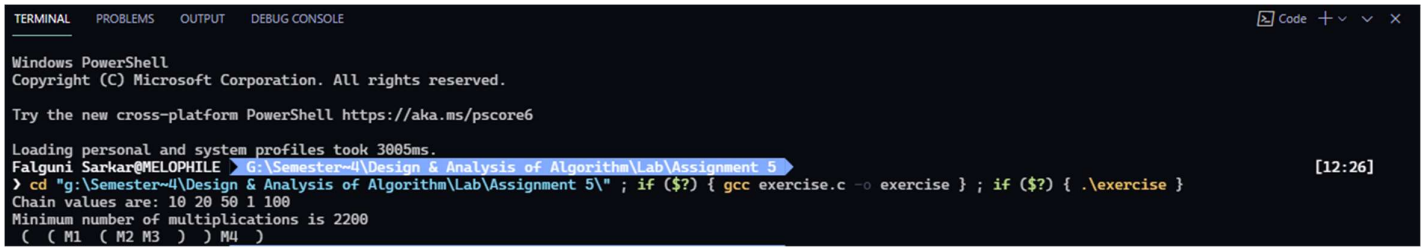
Procedure: matrixChainOrder(int a[], int n, int m[][n])
Inputs: s[][n] – array (or matrix) of size n.
a[n] – given array.
n – size of given array.
Output: It will return the minimum number of multiplications.

```
1. for i ← 1 to n do
2.         m[i][n] ← 0.
3. end for.
4. for l ← 2 to n do
5.         for i ← 1 to n – l + 1 do
6.                 j ← l + i – 1.
7.                 m[i][j] = INT_MAX;
8.                 for k ← i to j – 1 do
9.                         cost ← m[i][k] + m[k+1][j] + a[i-1] * a[j] * a[k];
10.                        if  cost < m[i][j] do
11.                                m[i][j] ← cost.
12.                                s[i][j] ← k.
13.                        end if.
14.                end for.
15.        end for.
16. end for.
17. return m[1][n-1].
```

Program –

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

void printOptimalParent(int s[][5], int i, int j)
{
    if (i == j)
        printf("M%d ", i);
    else
    {
        printf(" ( ");
        printOptimalParent(s, i, s[i][j]);
        printOptimalParent(s, s[i][j] + 1, j);
        printf(" ) ");
    }
}

int matrixChainOrder(int a[], int n, int s[][5])
{
    int i, j, k, l, cost;
    int m[n][n];

    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 = l + i - 1;
            m[i][j] = INT_MAX;
            for (k = i; k <= j - 1; k++)
            {
                cost = m[i][k] + m[k + 1][j] + a[i - 1] * a[j] * a[k];
                if (cost < m[i][j])
                {
                    m[i][j] = cost;
                    s[i][j] = k;
                }
            }
        }
    }
    return m[1][n - 1];
}

int main()
{

    int arr[] = {10, 20, 50, 1, 100};
    int i, n = 5;
    int s[5][5];

    printf("Chain values are: ");
    for (i = 0; i < 5; i++)
        printf("%d ", arr[i]);

    printf("\nMinimum number of multiplications is %d\n",
matrixChainOrder(arr, n, s));

    printOptimalParent(s, 1, n - 1);

    printf("\n");
    return 0;
}
```

Output –


TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
Loading personal and system profiles took 3005ms.
Falguni Sarkar@MELOPHILE G:\Semester~4\Design & Analysis of Algorithm\Lab\Assignment 5
[12:26]
cd "g:\Semester~4\Design & Analysis of Algorithm\Lab\Assignment 5\" ; if ($?) { gcc exercise.c -o exercise } ; if ($?) { .\exercise }
Chain values are: 10 20 50 1 100
Minimum number of multiplications is 2200
((M1 (M2 M3)) M4)