NAME:	Aditya Choudhary
UID:	2021300022
SUBJECT	Design and Analysis of Algorithm
EXPERIMENT NO :	04
DATE OF PERFORMANCE	14/03/2023
DATE OF SUBMISSION	21/03/2023
AIM:	To find the minimum matrix chain multiplications required.
PROBLEM STATEMENT 1:	**Matrix chain multiplication of matrices of different order.**
ALGORITHM and THEORY:	MATRIX-CHAIN-ORDER (p) 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.

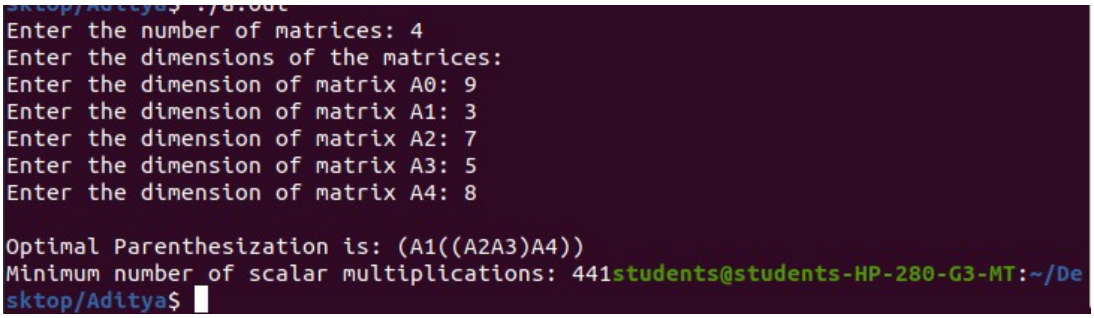
PROGRAM:

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

#define MAX 100

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

void printOptimalParentheses(int s[MAX][MAX], int i, int j) {
    if (i == j) {
        printf("A%d", i);
    } else {
        printf("(");
        printOptimalParentheses(s, i, s[i][j]);
        printOptimalParentheses(s, s[i][j] + 1, j);
        printf(")");
    }
```

```
}

int main() {
    int n, i, j;
    int p[MAX], m[MAX][MAX], s[MAX][MAX];

    printf("Enter the number of matrices: ");
    scanf("%d", &n);

    printf("Enter the dimensions of the matrices:\n");
    for (i = 0; i <= n; i++) {
        printf("Enter the dimension of matrix A%d: ", i);
        scanf("%d", &p[i]);
    }

    matrixChainOrder(p, n, m, s);

    printf("\nOptimal Parenthesization is: ");
    printOptimalParentheses(s, 1, n);

    printf("\nMinimum number of scalar multiplications: %d", m[1][n]);

    return 0;
}
```

OUTPUT:

```
students@students-HP-280-G3-MT:~/Desktop/Aditya$ gcc mc.c
students@students-HP-280-G3-MT:~/Desktop/Aditya$ ./a.out
Enter the number of matrices: 5
Enter the dimensions of the matrices:
Enter the dimension of matrix A0: 7
Enter the dimension of matrix A1: 4
Enter the dimension of matrix A2: 10
Enter the dimension of matrix A3: 6
Enter the dimension of matrix A4: 8
Enter the dimension of matrix A5: 11

Optimal Parenthesization is: (A1(((A2A3)A4)A5))
Minimum number of scalar multiplications: 1092students@students-HP-280-G3-MT:~/D
esktop/Aditya$
```

	
CONCLUSION:	I have successfully understood and implemented the concept of matrix chain multiplication through this experiment. I was also able to understand its uses and how dynamic programming reduces time complexity of matrix chain multiplication.