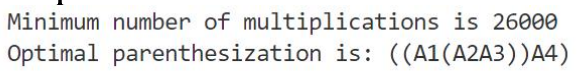
Question:

Matrix Chain Multiplication

Solution:

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

#define N 100

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

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

    printf("Minimum number of multiplications is %d\n", m[1][n-1]);
    printf("Optimal parenthesization is: ");
    printOptimalParens(s, 1, n-1);
    printf("\n");
}

int main() {
    int arr[] = {40, 20, 30, 10, 30};
    int n = sizeof(arr) / sizeof(arr[0]);

    matrixChainOrder(arr, n);

    return 0;
}
```

Output:--



```
Minimum number of multiplications is 26000
Optimal parenthesization is: ((A1(A2A3))A4)
```