

## DAA EXP 5

Name: Rahul Chalwadi

UID: 2021700012

CSE-DS(d1 batch)

**AIM:** To Implement the Matrix Chain Multiplication in C.

**Algorithm:**

### 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  $s$ .

CODE:

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

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

```
#define MAX_SIZE 100
```

```
//print the optimal eqn
```

```
void print_bracket(int s[MAX_SIZE][MAX_SIZE], int i, int j, char name) {  
    if (i == j) {  
        printf("%c", name++);  
    } else {  
        printf("(");  
        print_bracket(s, i, s[i][j], name);  
        print_bracket(s, s[i][j]+1, j, name+s[i][j]-i+1);  
        printf(")");  
    }  
}
```

```
//minimum cost
```

```

int mcc(int p[], int n, char name) {
    int m[MAX_SIZE][MAX_SIZE], s[MAX_SIZE][MAX_SIZE];

    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("optimal eqn of multiplication of matrix : ");
    print_bracket(s, 1, n, name);
    printf("\n");

    return m[1][n];
}

int main() {
    int num_matrices;
    printf("Enter the number of matrices: ");
    scanf("%d", &num_matrices);

    int matrices[num_matrices][2]; //cannot compute more than 2 dimensions matrix

    for (int i = 0; i < num_matrices; i++) {
        printf("Enter the dimensions of matrix %c: ", 'A' + i);
        scanf("%d %d", &matrices[i][0], &matrices[i][1]);
    }

    int matrix_sizes[MAX_SIZE];
    int idx = 0;

```

```

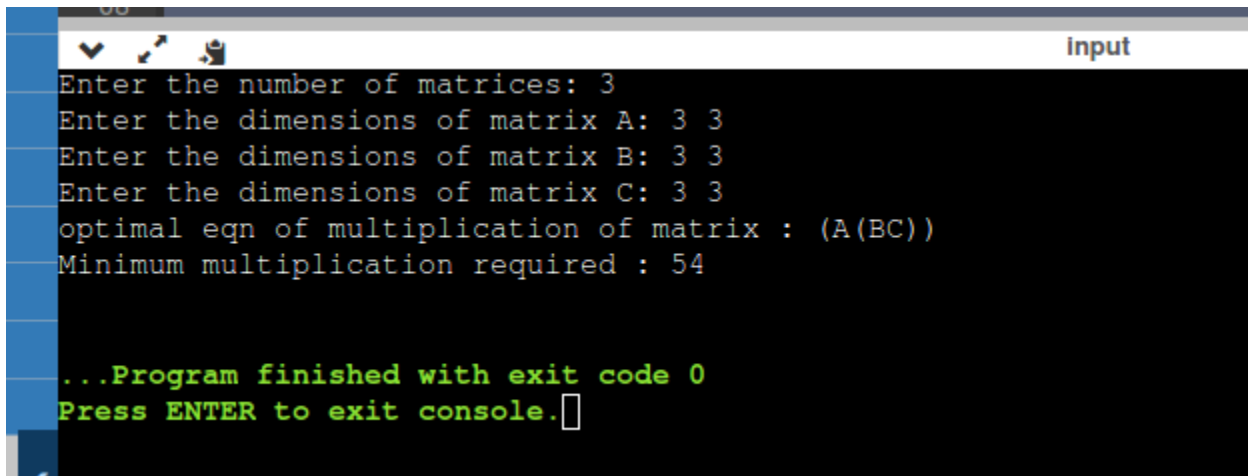
    for (int i = 0; i < num_matrices; i++) {
        matrix_sizes[idx++] = matrices[i][0];
        if (i == num_matrices - 1) {
            matrix_sizes[idx++] = matrices[i][1];
        }
    }

    printf("Minimum multiplication required : %d\n", mcc(matrix_sizes, idx - 1, 'A'));

    return 0;
}

```

OUTPUT:



```

Input
Enter the number of matrices: 3
Enter the dimensions of matrix A: 3 3
Enter the dimensions of matrix B: 3 3
Enter the dimensions of matrix C: 3 3
optimal eqn of multiplication of matrix : (A(BC))
Minimum multiplication required : 54

...Program finished with exit code 0
Press ENTER to exit console.

```

Conclusion:

IN this experiment, I understood about matrix chain multiplication and implemented it in the C language.