Assignment- II

Name: Soumyadip Ghosh

Stream: CSE- A

Roll Number: 1951007

Subject: Algorithms Lab

Problem:

Matrix Chain Multiplication
1. Input should be taken from Keyboard
2. Print the lower triangular matrix of m and s
3. Print the optimal cost
4. Print the parenthesized Matrices.

Solution:

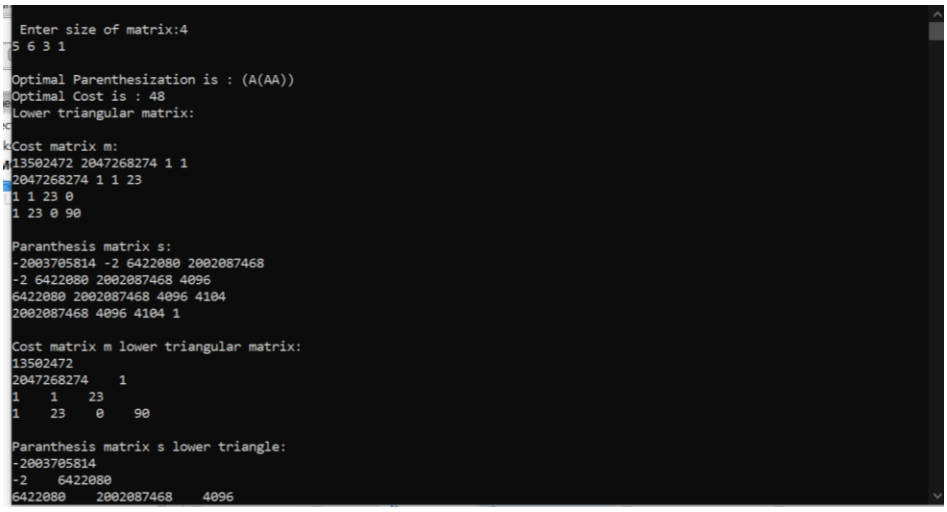
```
#include <stdio.h>

#include <stdlib.h>

#include <limits.h>





void mcm(int p[], int n);

void parenthesizedMatrix(int i,int j,int n, int* s,char name);

void printMatrix(int* m,int* s,int n);


int main()

{

    int n,i;

    printf("\n Enter size of matrix:");

    scanf("%d",&n);

    int arr[n];

    for(i=0;i<n;i++)

    scanf("%d",&arr[i]);

    mcm(arr,n);


    return 0;

}


void mcm(int p[], int n)
```

```
{

    int i,j,l,k,a=0,cost=0;
    int m[n][n],s[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= i+l-1;
            m[i][j] = INT_MAX;
            for (k=i; k<=j-1; k++)
            {
                cost= m[i][k] + m[k + 1][j]+ p[i - 1]*p[k]*p[j];
                if (cost< m[i][j])
                {
                    m[i][j] = cost;
                    s[i][j] = k;
                }
            }
        }
    }

    char name = 'A';

    printf("\nOptimal Parenthesization is : ");
    parenthesizedMatrix(1,n - 1,n,(int*)s, name);
    printf("\nOptimal Cost is : %d" ,m[1][n - 1]);
    printf("\nLower triangular matrix:\n");
    printMatrix((int*)m,(int*)s,n);
}

void parenthesizedMatrix(int i, int j, int n, int* s, char name)
{
    if (i == j) {
        printf("%c",name);
        return;
    }

    printf("(");
    parenthesizedMatrix(i,*((s+i*n)+j),n,s,name);
    parenthesizedMatrix(*((s+i*n)+j)+1,j,n,s,name);
    printf(")");
}

void printMatrix(int* m,int* s,int n){
    int i,j;
    printf("\nCost matrix m:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("%d ",*((m+i)+j));
        }
        printf("\n");
    }
    printf("\nParanthesis matrix s:\n");
    for(i=0;i<n;i++){
```

```
    for(j=0;j<n;j++){
        printf("%d ",*((s+i)+j));
    }
    printf("\n");
}
printf("\nCost matrix m lower triangular matrix:\n");
for (i=0; i<n;i++)
{
    for (j=0;j<n;j++)
    {
        if (i < j)
        {
        }
        else
        printf("%d    ",*((m+i)+j));
    }
    printf("\n");
}
printf("\nParanthesis matrix s lower triangle:\n");
 for (i=0; i<n;i++)
{
    for (j=0;j<n;j++)
    {
        if (i < j)
        {


        }
        else
        printf("%d    ",*((s+i)+j));
    }
```

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

Output:



```
 Enter size of matrix:4
5 6 3 1

Optimal Parenthesization is : (A(AA))
Optimal Cost is : 48
Lower triangular matrix:

Cost matrix m:
13502472 2047268274 1 1
2047268274 1 1 23
1 1 23 0
1 23 0 90

Paranthesis matrix s:
-2003705814 -2 6422080 2002087468
-2 6422080 2002087468 4096
6422080 2002087468 4096 4104
2002087468 4096 4104 1

Cost matrix m lower triangular matrix:
13502472
2047268274    1
1    1    23
1    23    0    90

Paranthesis matrix s lower triangle:
-2003705814
-2    6422080
6422080    2002087468    4096
```