

<b>Name</b>	Sujal Chordia
<b>UID no.</b>	2021700015
<b>Experiment No.</b>	5

<b>AIM:</b>	Dynamic Programming - Matrix Chain Multiplication.
<b>Program</b>	
<b>PROBLEM STATEMENT:</b>	Use Dynamic Programming method to find the optimal way to multiply(parenthesize) the matrices to find the minimum number of multiplications required to solve the matrix.
<b>ALGORITHM/ THEORY:</b>	<p><b>Matrix chain multiplication :</b></p> <p>Is an <a href="#">optimization problem</a> concerning the most efficient way to <a href="#">multiply</a> a given sequence of <a href="#">matrices</a>. The problem is not actually to <i>perform</i> the multiplications, but merely to decide the sequence of the matrix multiplications involved. The problem may be solved using <a href="#">dynamic programming</a>.</p> <p>There are many options because matrix multiplication is <a href="#">associative</a>. In other words, no matter how the product is <a href="#">parenthesized</a>, the result obtained will remain the same. For example, for four matrices <i>A</i>, <i>B</i>, <i>C</i>, and <i>D</i>, there are five possible options:</p> $((AB)C)D = (A(BC))D = (AB)(CD) = A((BC)D) = A(B(CD)).$
<b>PROGRAM:</b>	<pre>#include&lt;stdio.h&gt; #include&lt;time.h&gt; int mat[100][100],s[100][100],count=0; int MCM(int p[], int i, int j){     if(i==j){         mat[i][j] = 0;         return 0;     }     mat[i][j] = 30000;     for(int k=i; k&lt;j; k++){         count = MCM(p,i,k) + MCM(p,k+1,j) + p[i-1]*p[k]*p[j];     } }</pre>

```

        if(count<mat[i][j]){
            mat[i][j] = count;
            s[i][j] = k;
        }
    }
    return mat[i][j];
}

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

void main(){
    int num;
    printf("\n The number of inputs : ");
    scanf("%d",&num);
    int p[num];
    // printf("\nEnter the order of matrices: ");
    for(int i=0;i<num;i++){
        printf("\n The value for place is %d: ",i+1);
        scanf("%d",&p[i]);
    }
    printf("\n Minimum number of multiplications : %d\n\n",MCM(p,1,num-1));
    for(int i=1;i<num;i++){
        for(int j=1;j<num;j++){
            printf("%d\t",mat[i][j]);
        }
        printf("\n");
    }
    printf("\n The optimal solution : \n");
    POP(1,num-1);
}

```

**RESULT:**

✓ ↗ 📄

The number of inputs : 4

The value for place is 1: 10

The value for place is 2: 20

The value for place is 3: 30

The value for place is 4: 40

< Minimum number of multiplications : 18000

0	6000	18000
0	0	24000
0	0	0

The optimal solution :  
((S1S2)S3)

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

<b>CONCLUSION:</b>	The Dynamic Programming steps was used to solve the Matrix Chain Multiplication problem.
--------------------	--