Shubhan Singh

SE-Comps B/Batch C

2022300118

DAA Experiment 5 : Matrix Chain Multiplication

Aim – To implement Dynamic programming algorithms to find the optimal order for Matrix chain multiplication.

Details – Dynamic Programming is a technique in computer programming that helps to efficiently solve a class of problems that have overlapping sub-problems and optimal substructure property. If any problem can be divided into sub-problems, which in turn are divided into smaller sub-problems, and if there are overlapping among these subproblems, then the solutions to these sub-problems can be saved for future reference.

Problem statement:

Consider the optimization problem of efficiently multiplying a sequence of n matrices (M1, M2, M3, M4 ,..., Mn) using Dynamic programming approach. The dimension of these matrices are stored in an array p[i] for i = 0 to n, where the dimension of the matrix Mi is (p[i-1] x p[i]).

Determine following values of Matrix Chain Multiplication (MCM) using Dynamic Programming:
1) m[1..n][1..n] = Two dimension matrix of optimal solutions (No. of multiplications) of all possible matrices M1... Mn

2) the optimal solution (i.e.parenthesization) for the multiplication of all n matrices M1x M2x M3xM4 x...x Mn

Pseudocode :

```
Algorithm Matrix-Chain-Multiplication(p, n)
  Input: Sequence p[0..n] of matrix dimensions, number of matrices n
  Output: The minimum number of scalar multiplications needed to compute the product of the matrices


  1. Let m[1..n, 1..n] and s[1..n, 1..n] be new tables
```

```
for i = 1 to n
    m[i, i] = 0
for l = 2 to n    // l is the chain length
    for i = 1 to n-l+1
        j = i+l-1
        m[i, j] = infinity
        for k = i to j-1
            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
Print the matrix of costs m
Print the matrix s
Call the function Parenthesize(1, n, s) to print the optimal parenthesization
return m[1, n]
```

M[1,n] contains the minimum number of scalar multiplications required

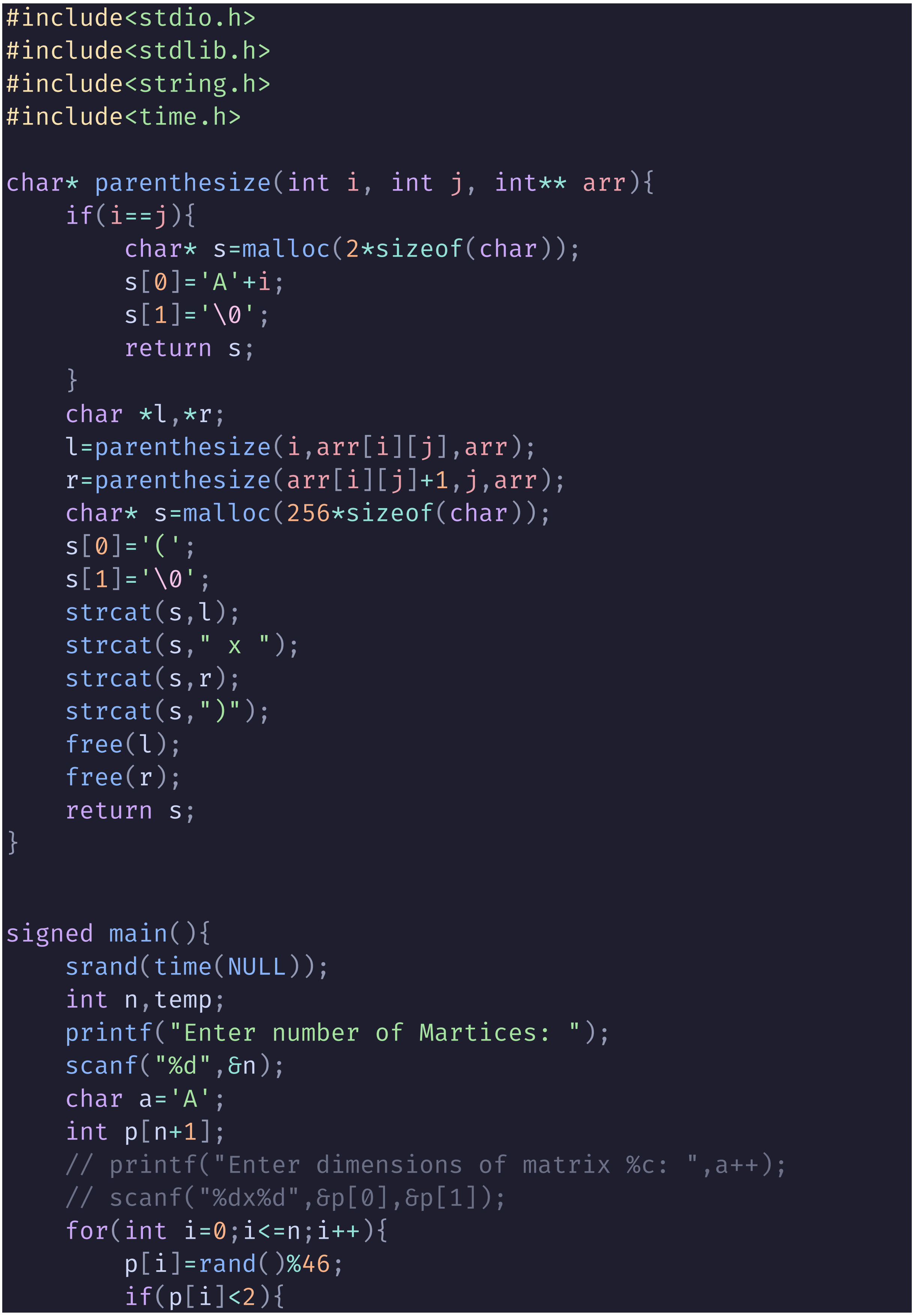
Function Parenthesize(i, j, s)

Input: Matrix s of splitting points, indices i and j

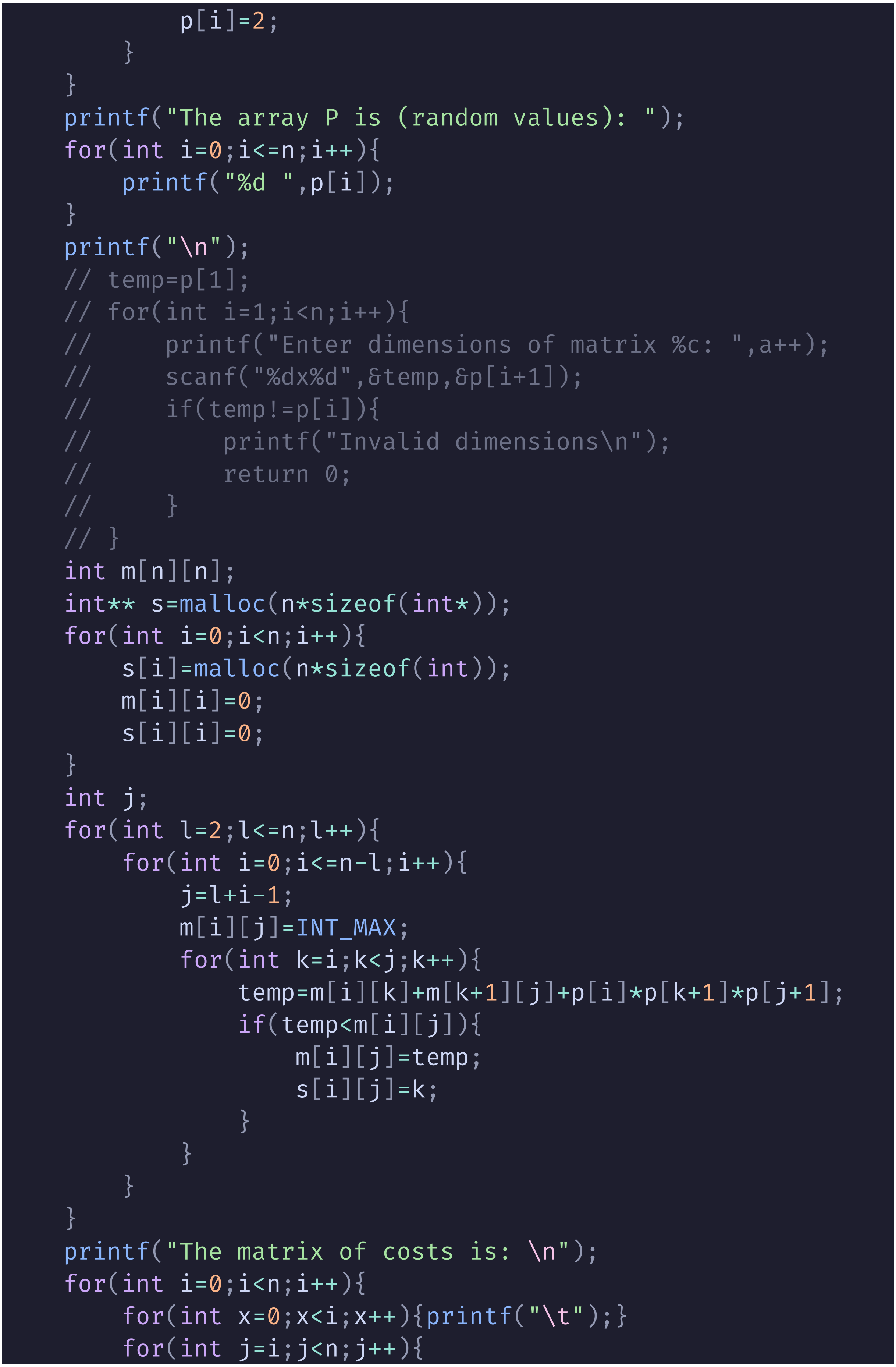
Output: A string representing the optimal parenthesization

```
if i == j
    return "A" + i
else
    return "(" + Parenthesize(i, s[i, j], s) + " x " + Parenthesize(s[i, j] + 1, j, s) + ")"
```

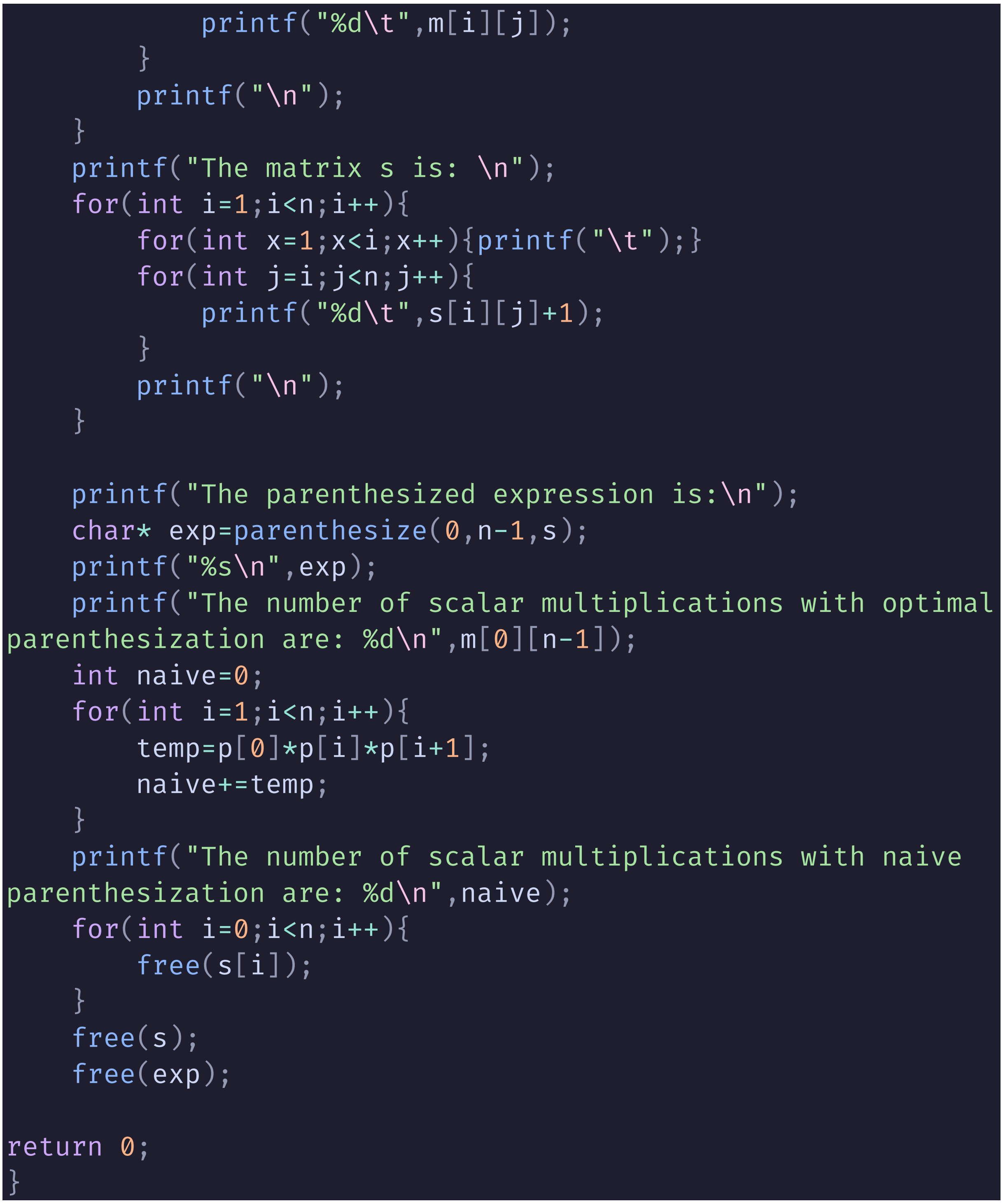
Source code(C language):



```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>

char* parenthesize(int i, int j, int** arr){
    if(i==j){
        char* s=malloc(2*sizeof(char));
        s[0]='A'+i;
        s[1]='\0';
        return s;
    }
    char *l,*r;
    l=parenthesize(i,arr[i][j],arr);
    r=parenthesize(arr[i][j]+1,j,arr);
    char* s=malloc(256*sizeof(char));
    s[0]='(';
    s[1]='\0';
    strcat(s,l);
    strcat(s," x ");
    strcat(s,r);
    strcat(s,")");
    free(l);
    free(r);
    return s;
}


signed main(){
    srand(time(NULL));
    int n,temp;
    printf("Enter number of Martices: ");
    scanf("%d",&n);
    char a='A';
    int p[n+1];
    // printf("Enter dimensions of matrix %c: ",a++);
    // scanf("%dx%d",&p[0],&p[1]);
    for(int i=0;i<=n;i++){
        p[i]=rand()%46;
        if(p[i]<2){
```



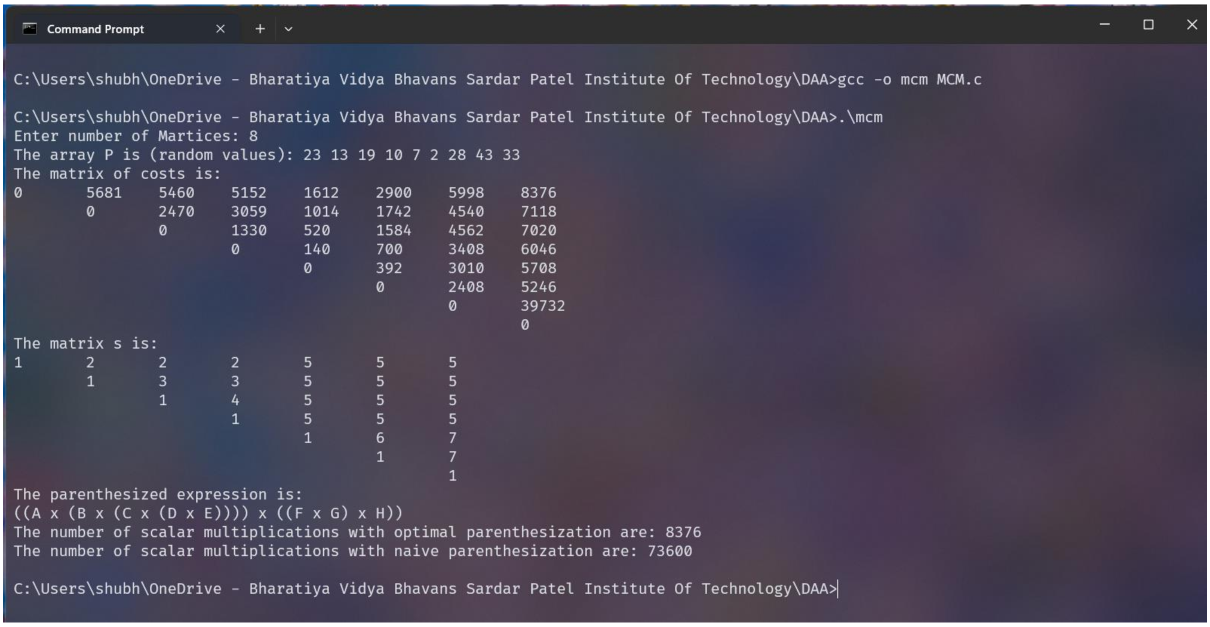
```
            p[i]=2;
        }
    }
    printf("The array P is (random values): ");
    for(int i=0;i<=n;i++){
        printf("%d ",p[i]);
    }
    printf("\n");
    // temp=p[1];
    // for(int i=1;i<n;i++){
    //     printf("Enter dimensions of matrix %c: ",a++);
    //     scanf("%dx%d",&temp,&p[i+1]);
    //     if(temp!=p[i]){
    //         printf("Invalid dimensions\n");
    //         return 0;
    //     }
    // }
    int m[n][n];
    int** s=malloc(n*sizeof(int*));
    for(int i=0;i<n;i++){
        s[i]=malloc(n*sizeof(int));
        m[i][i]=0;
        s[i][i]=0;
    }
    int j;
    for(int l=2;l<=n;l++){
        for(int i=0;i<=n-l;i++){
            j=l+i-1;
            m[i][j]=INT_MAX;
            for(int k=i;k<j;k++){
                temp=m[i][k]+m[k+1][j]+p[i]*p[k+1]*p[j+1];
                if(temp<m[i][j]){
                    m[i][j]=temp;
                    s[i][j]=k;
                }
            }
        }
    }
    printf("The matrix of costs is: \n");
    for(int i=0;i<n;i++){
        for(int x=0;x<i;x++){printf("\t");}
        for(int j=i;j<n;j++){
```



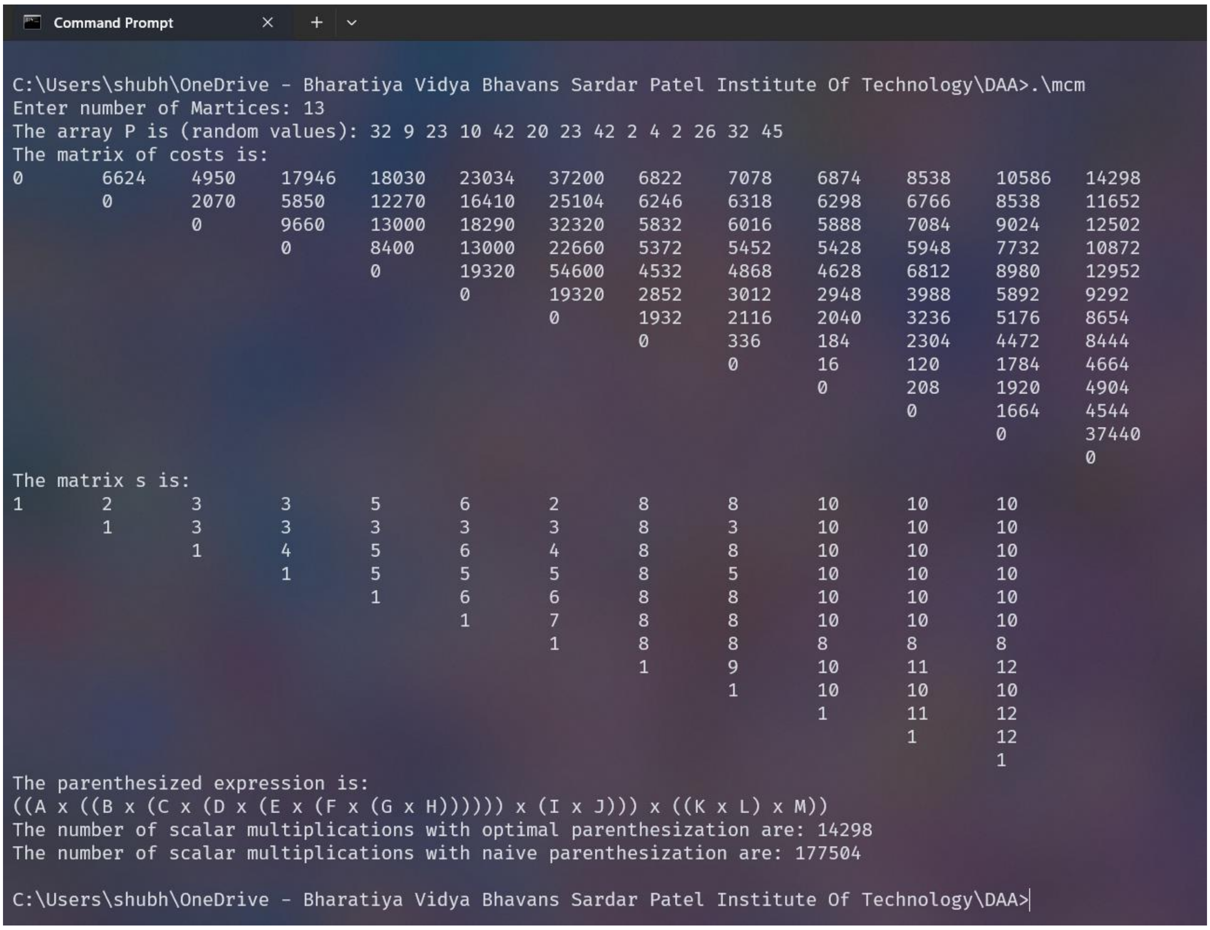
```c
            printf("%d\t",m[i][j]);
        }
        printf("\n");
    }
    printf("The matrix s is: \n");
    for(int i=1;i<n;i++){
        for(int x=1;x<i;x++){printf("\t");}
        for(int j=i;j<n;j++){
            printf("%d\t",s[i][j]+1);
        }
        printf("\n");
    }

    printf("The parenthesized expression is:\n");
    char* exp=parenthesize(0,n-1,s);
    printf("%s\n",exp);
    printf("The number of scalar multiplications with optimal
parenthesization are: %d\n",m[0][n-1]);
    int naive=0;
    for(int i=1;i<n;i++){
        temp=p[0]*p[i]*p[i+1];
        naive+=temp;
    }
    printf("The number of scalar multiplications with naive
parenthesization are: %d\n",naive);
    for(int i=0;i<n;i++){
        free(s[i]);
    }
    free(s);
    free(exp);

return 0;
}
```

Output:

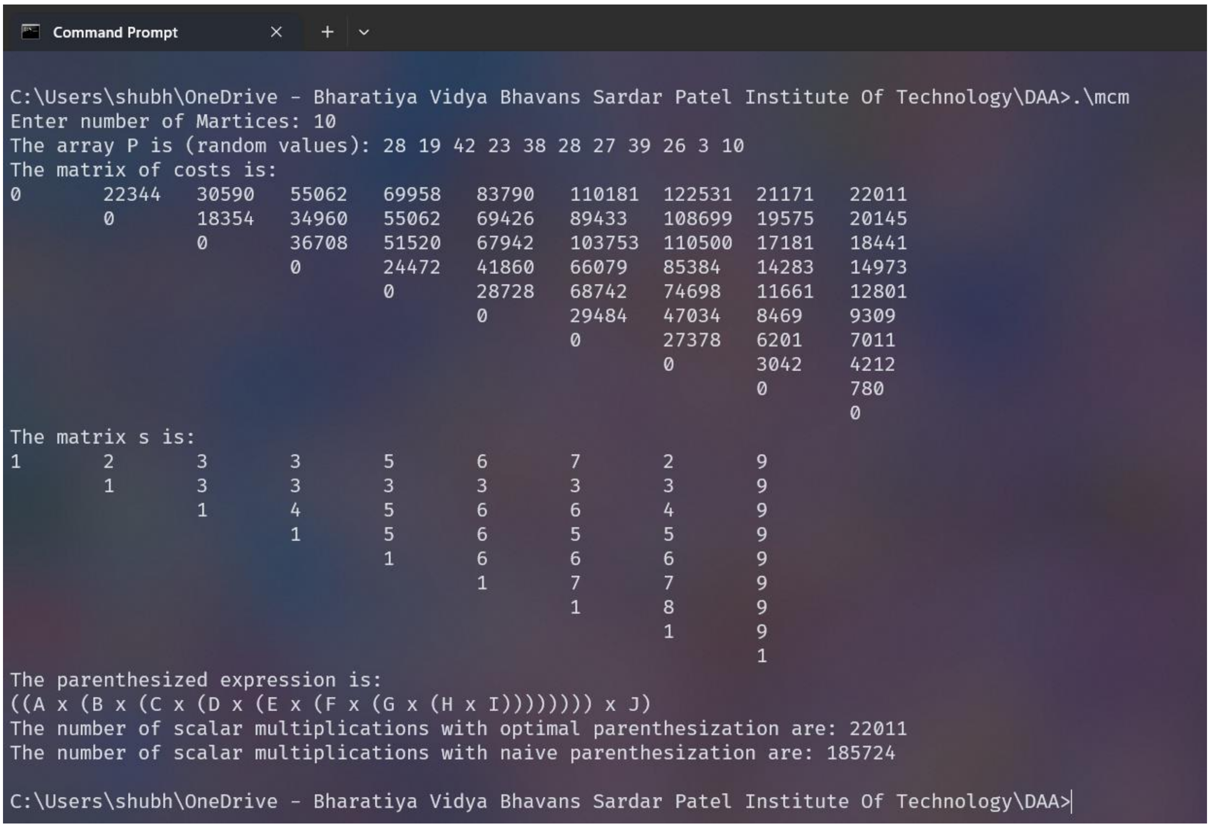
(5 cases)



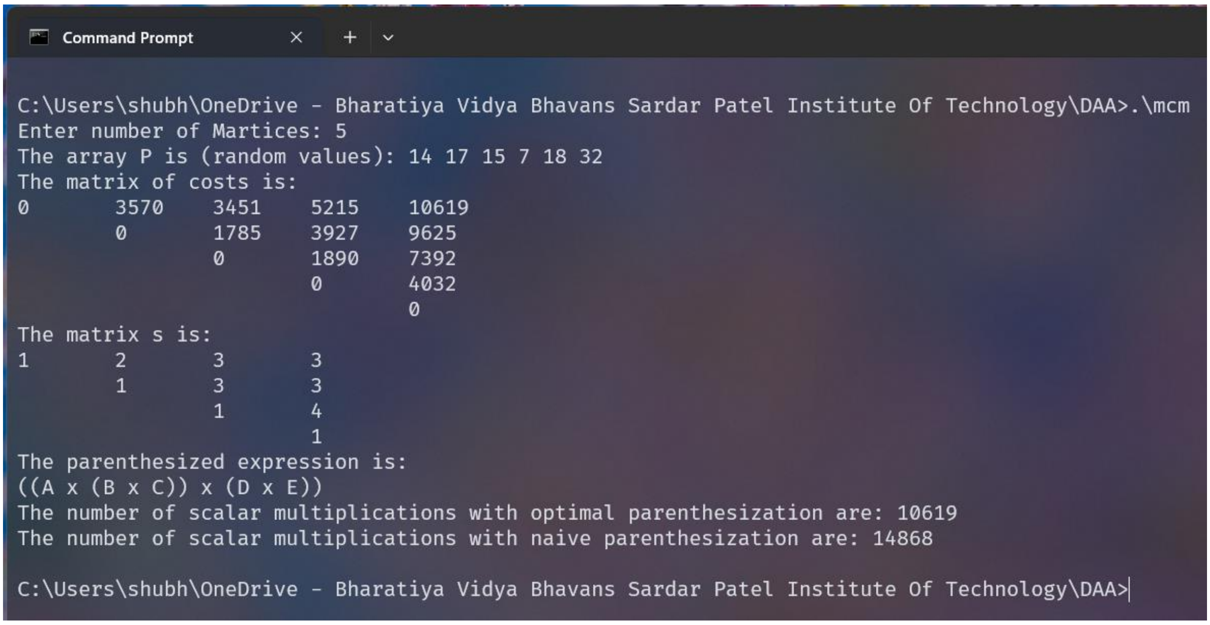
```
C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>gcc -o mcm MCM.c

C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>.\mcm
Enter number of Martices: 8
The array P is (random values): 23 13 19 10 7 2 28 43 33
The matrix of costs is:
0       5681    5460    5152    1612    2900    5998    8376
        0       2470    3059    1014    1742    4540    7118
                0       1330    520     1584    4562    7020
                        0       140     700     3408    6046
                                0       392     3010    5708
                                        0       2408    5246
                                                0       39732
                                                        0
The matrix s is:
1       2       2       2       5       5       5
        1       3       3       5       5       5
                1       4       5       5       5
                        1       5       5       5
                                1       6       7
                                        1       7
                                                1
The parenthesized expression is:
((A x (B x (C x (D x E)))) x ((F x G) x H))
The number of scalar multiplications with optimal parenthesization are: 8376
The number of scalar multiplications with naive parenthesization are: 73600

C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>
```



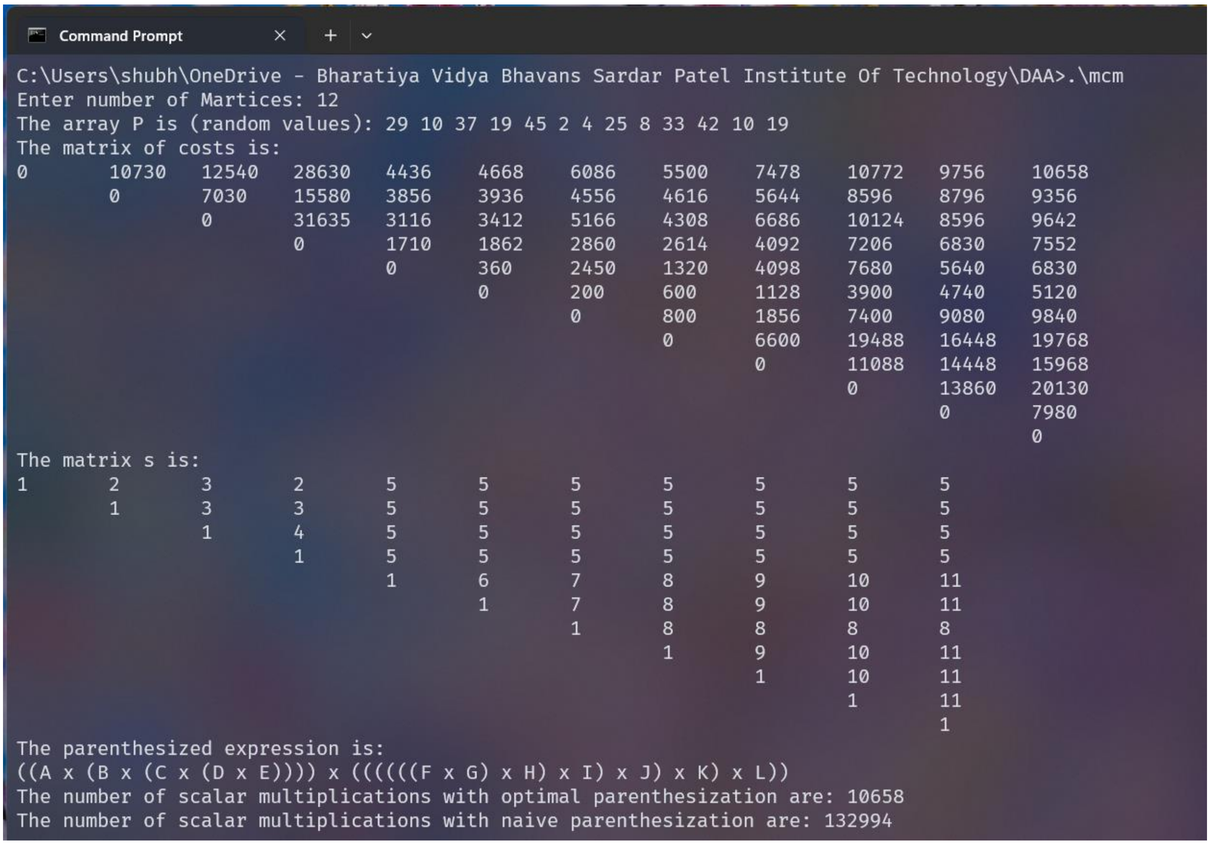
```
C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>.\mcm
Enter number of Martices: 13
The array P is (random values): 32 9 23 10 42 20 23 42 2 4 2 26 32 45
The matrix of costs is:
0       6624    4950    17946   18030   23034   37200   6822    7078    6874    8538    10586   14298
        0       2070    5850    12270   16410   25104   6246    6318    6298    6766    8538    11652
                0       9660    13000   18290   32320   5832    6016    5888    7084    9024    12502
                        0       8400    13000   22660   5372    5452    5428    5948    7732    10872
                                0       19320   54600   4532    4868    4628    6812    8980    12952
                                        0       19320   2852    3012    2948    3988    5892    9292
                                                0       1932    2116    2040    3236    5176    8654
                                                        0       336     184     2304    4472    8444
                                                                0       16      120     1784    4664
                                                                        0       208     1920    4904
                                                                                0       1664    4544
                                                                                        0       37440
                                                                                                0
The matrix s is:
1       2       3       3       5       6       2       8       8       10      10      10
        1       3       3       3       3       3       8       3       10      10      10
                1       4       5       6       4       8       8       10      10      10
                        1       5       5       5       8       5       10      10      10
                                1       6       6       8       8       10      10      10
                                        1       7       8       8       10      10      10
                                                1       8       8       8       8       8
                                                        1       9       10      11      12
                                                                1       10      10      10
                                                                        1       11      12
                                                                                1       12
                                                                                        1
The parenthesized expression is:
((A x ((B x (C x (D x (E x (F x (G x H)))))) x (I x J))) x ((K x L) x M))
The number of scalar multiplications with optimal parenthesization are: 14298
The number of scalar multiplications with naive parenthesization are: 177504

C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>
```



```
C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>.\mcm
Enter number of Martices: 10
The array P is (random values): 28 19 42 23 38 28 27 39 26 3 10
The matrix of costs is:
0       22344   30590   55062   69958   83790   110181  122531  21171   22011
        0       18354   34960   55062   69426   89433   108699  19575   20145
                0       36708   51520   67942   103753  110500  17181   18441
                        0       24472   41860   66079   85384   14283   14973
                                0       28728   68742   74698   11661   12801
                                        0       29484   47034   8469    9309
                                                0       27378   6201    7011
                                                        0       3042    4212
                                                                0       780
                                                                        0
The matrix s is:
1       2       3       3       5       6       7       2       9
        1       3       3       3       3       3       3       9
                1       4       5       6       6       4       9
                        1       5       6       5       5       9
                                1       6       6       6       9
                                        1       7       7       9
                                                1       8       9
                                                        1       9
                                                                1
The parenthesized expression is:
((A x (B x (C x (D x (E x (F x (G x (H x I)))))))) x J)
The number of scalar multiplications with optimal parenthesization are: 22011
The number of scalar multiplications with naive parenthesization are: 185724

C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>
```



```
C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>.\mcm
Enter number of Martices: 5
The array P is (random values): 14 17 15 7 18 32
The matrix of costs is:
0       3570    3451    5215    10619
        0       1785    3927    9625
                0       1890    7392
                        0       4032
                                0
The matrix s is:
1       2       3       3
        1       3       3
                1       4
                        1
The parenthesized expression is:
((A x (B x C)) x (D x E))
The number of scalar multiplications with optimal parenthesization are: 10619
The number of scalar multiplications with naive parenthesization are: 14868

C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>
```

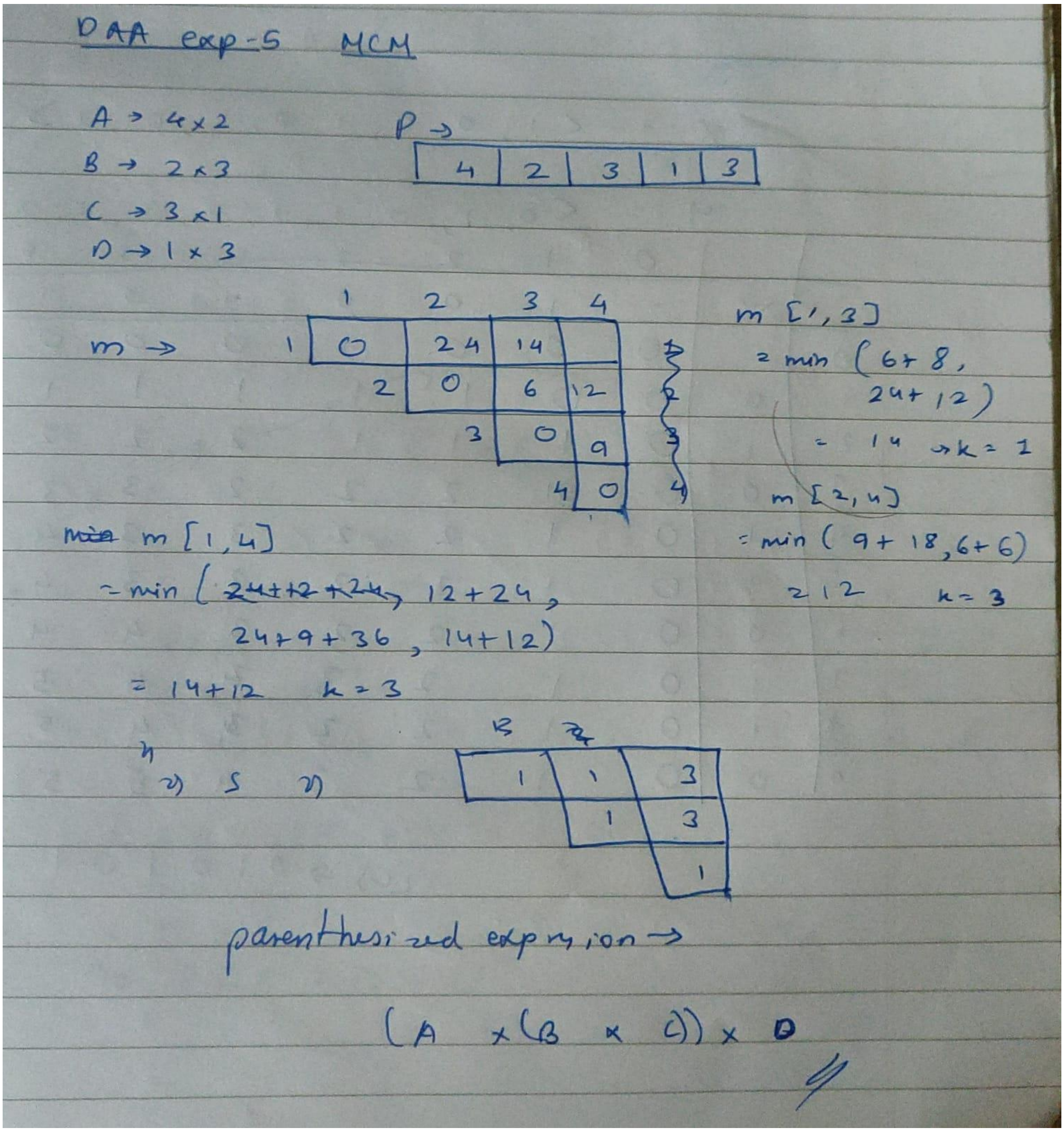


```
C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\DAA>.\mcm
Enter number of Martices: 12
The array P is (random values): 29 10 37 19 45 2 4 25 8 33 42 10 19
The matrix of costs is:
0       10730   12540   28630   4436    4668    6086    5500    7478    10772   9756    10658
        0       7030    15580   3856    3936    4556    4616    5644    8596    8796    9356
                0       31635   3116    3412    5166    4308    6686    10124   8596    9642
                        0       1710    1862    2860    2614    4092    7206    6830    7552
                                0       360     2450    1320    4098    7680    5640    6830
                                        0       200     600     1128    3900    4740    5120
                                                0       800     1856    7400    9080    9840
                                                        0       6600    19488   16448   19768
                                                                0       11088   14448   15968
                                                                        0       13860   20130
                                                                                0       7980
                                                                                        0
The matrix s is:
1       2       3       2       5       5       5       5       5       5       5
        1       3       3       5       5       5       5       5       5       5
                1       4       5       5       5       5       5       5       5
                        1       5       5       5       5       5       5       5
                                1       6       7       8       9       10      11
                                        1       7       8       9       10      11
                                                1       8       8       8       8
                                                        1       9       10      11
                                                                1       10      11
                                                                        1       11
                                                                                1
The parenthesized expression is:
((A x (B x (C x (D x E)))) x ((((((F x G) x H) x I) x J) x K) x L))
The number of scalar multiplications with optimal parenthesization are: 10658
The number of scalar multiplications with naive parenthesization are: 132994
```

Conclusion:

- This dynamic programming solution executes in O(n^3) time complexity, where n is the number of matrices we are trying to multiply.
- This solution satisfies the optimal substructure property as in the process of arriving to the final answer, we also found the optimal solutions to all of the all of the subproblems (the minimum number of scalar multiplications required to multiply Matrices i…j).

Rough Working :



DAA exp-5 MCM

A → 4×2

B → 2×3

C → 3×1

D → 1×3

P →

4	2	3	1	3

m →

	1	2	3	4
1	0	24	14	
2		0	6	12
3			0	9
4				0

m [1, 3]
= min (6+8,
24+12)
= 14 → k = 1

m [2, 4]
= min (9+18, 6+6)
= 12 k = 3

~~min~~ m [1, 4]
= min (~~24+12+24~~, 12+24,
24+9+36, 14+12)
= 14+12 k = 3

2) S 2)

B	Z	
1	1	3
	1	3
		1

parenthesized expression →

(A × (B × C)) × D

Theory:

1. **Problem Statement**: The Matrix Chain Multiplication problem is an optimization problem that deals with the most efficient way to multiply a chain of matrices. The problem is not to perform the multiplications, but merely to decide the sequence of the matrix multiplications involved.
2. **Order Matters**: The order of matrix multiplication matters because the cost of multiplication can vary dramatically depending on the order. For example, if you have three matrices A, B, and C with dimensions 10x100, 100x5, and 5x50 respectively, then (A(BC)) would require 7500 scalar multiplications, while ((AB)C) would require 25000.
3. **Dynamic Programming Solution**: The problem can be solved using dynamic programming by breaking it down into smaller subproblems, solving each subproblem only once, and storing their results in case they are needed later (this is known as memoization).
4. **Subproblems**: The subproblems are defined by a starting and ending position for the chain of matrices to be multiplied (i.e., for each pair (i, j) where $1 \leq i \leq j \leq n$, find the most efficient way to multiply matrices i through j in the chain).
5. **Recurrence Relation**: The dynamic programming solution uses a recurrence relation to express the solution of the problem in terms of smaller subproblems. The minimum number of multiplications needed to multiply matrices i through j is found by trying all possibilities for the final multiplication, and choosing the one that costs the least.
6. **Parenthesization**: After the table is filled, the solution to the problem can be found by tracing back through the decisions that led to the optimal cost. This gives the optimal parenthesization of the matrix chain.
7. **Time Complexity**: The time complexity of the dynamic programming solution to the Matrix Chain Multiplication problem is O(n^3), where n is the number of matrices in the chain.