NAME:	Deepanshu Aggarwal
UID:	2021300002
BRANCH:	Computer engineering
BATCH:	A
SUBJECT:	DAA
EXPT NO:	4

AIM:	Experiment using dynamic programming approach (Implementation of Matrix Chain Multiplication)
ALGORITHM/ THEORY:	1. n length[p]-1 2. for i ← 1 to n 3. do m [i, i] ← 0 4. for l ← 2 to n // l is the chain length 5. do for i ← 1 to n-l + 1 6. do j ← i+ l -1 7. m[i,j] ← ∞ 8. for k ← i to j-1 9. do q ← 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] ← q 12. s [i,j] ← k 13. return m and s.

PROGRAM:	`#include<stdio.h>`
	`#include<stdlib.h>`
	`#include<limits.h>`
	
	`int** carr, **karr;`
	`void matrixchain( int arr[],int n){`

```
    int tempk;
    for(int l=2;l<n;l++){
        for(int i=1;i<n-l+1;i++){
            int j = i+l-1;
            carr[i][j]=INT_MAX;
            for(int k=i;k<j;k++){
               int cal = carr[i][k] + carr[k+1][j] + arr[i-
1]*arr[j]*arr[k];
               if(carr[i][j]>cal){
                  carr[i][j]= cal;
                  karr[i][j]= k;
               }
            }

        }
    }
    printf("\nThe resultant matrix is: \n");
    for(int i=1;i<n-1;i++){
        for(int j=2;j<n;j++){
            printf("%d\t",carr[i][j]);
        }
        printf("\n");
    }
    printf("\nResultant k matrix is: \n");
    for(int i=1;i<n-1;i++){
        for(int j=2;j<n;j++){
            printf("%d  ",karr[i][j]);
        }
        printf("\n");
    }
}
int main(){
    int n;
    printf("Enter the number of matrix: ");
    scanf("%d",&n);
    n=n+1;
    int arr[n];
    printf("Enter the corresponding dimensions of the
Matrices: ");
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    for(int i=0;i<n;i++){
        printf("%d  ",arr[i]);
```

```
    }
    carr=(int**)calloc(n,sizeof(int*));
    for(int i=0; i<n; i++){
        carr[i]=(int*)calloc(n,sizeof(int));
    }

    karr=(int**)calloc(n,sizeof(int*));
    for(int i=0; i<n; i++){
        karr[i]=(int*)calloc(n,sizeof(int));
    }

    matrixchain(arr,n);
}
```

RESULT:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE

Enter the number of matrix: 4
Enter the corresponding dimensions of the Matrices: 40 20 30 10 30
40  20  30  10  30
The resultant matrix is:
24000   14000   26000
0       6000    12000
0       0       9000

Resultant k matrix is:
1  1  3
0  2  3
0  0  3

Minimum number of multiplications is 26000
```

CONCLUSION: Through this experiment I learnt the concept of Dynamic Programming. Also, I learnt and implemented its one of the algorithms i.e., Matrix chain multiplication. It helped me to find minimum number of multiplications when there are multiple matrices.