**Agnim Gupta**

**2028083**

**A-23,CSSE**

**Question 1**

#include <limits.h>

#include <stdio.h>

int MatrixChainOrder(int p[], int n)

{

int m[n][n];

int i, j, k, L, q;

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++) {

q = m[i][k] + m[k + 1][j] + p[i - 1] \* p[k] \* p[j]; if (q < m[i][j])

m[i][j] = q;

}

}

}

return m[1][n - 1];

}

int main()

{

int arr[] = { 1, 2, 3, 4 };

int size = sizeof(arr) / sizeof(arr[0]);

printf("Minimum number of multiplications is %d ", MatrixChainOrder(arr, size));

getchar();

return 0;

}

**Output**



**Question 2**

#include<stdio.h>

#include<limits.h>

int MatrixChainMultiplication(int p[], int n)

{

    int m[n][n];

    int i, j, k, L, q;

    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++)

            {

                q = m[i][k] + m[k+1][j] + p[i-1]\*p[k]\*p[j];

                if (q < m[i][j])

                {

                    m[i][j] = q;

                }

            }

        }

    }

    return m[1][n-1];

}

int main()

{

    int n,i;

    printf("Enter number of matrices\n");

    scanf("%d",&n);

    n++;

    int arr[n];

    printf("Enter dimensions \n");

    for(i=0;i<n;i++)

    {

        printf("Enter d%d :: ",i);

        scanf("%d",&arr[i]);

    }

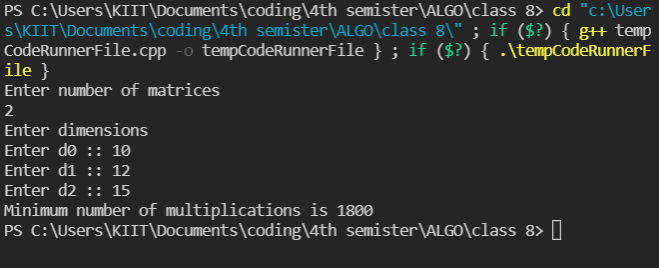
    int size = sizeof(arr)/sizeof(arr[0]);

    printf("Minimum number of multiplications is %d ", MatrixChainMultiplication(arr, size));

    return 0;

}

**Output**

****