NAME – ANIKET SHARMA

REG. NO – RA1911003030436

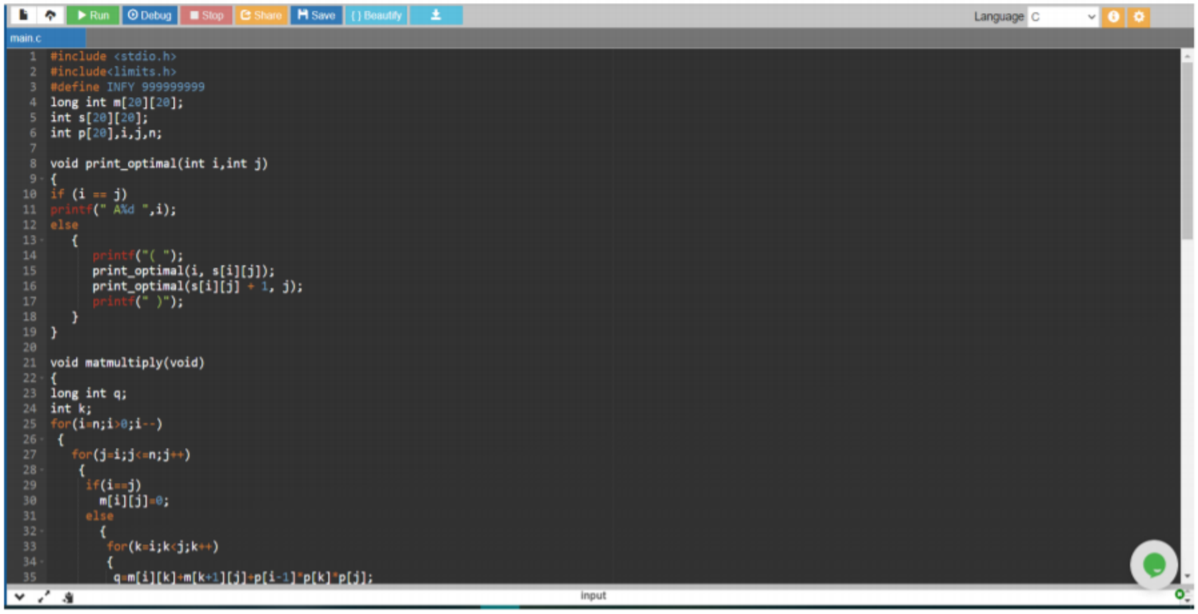
BRANCH- CSE-I

SUBJECT- Design and Analysis of Algorithms

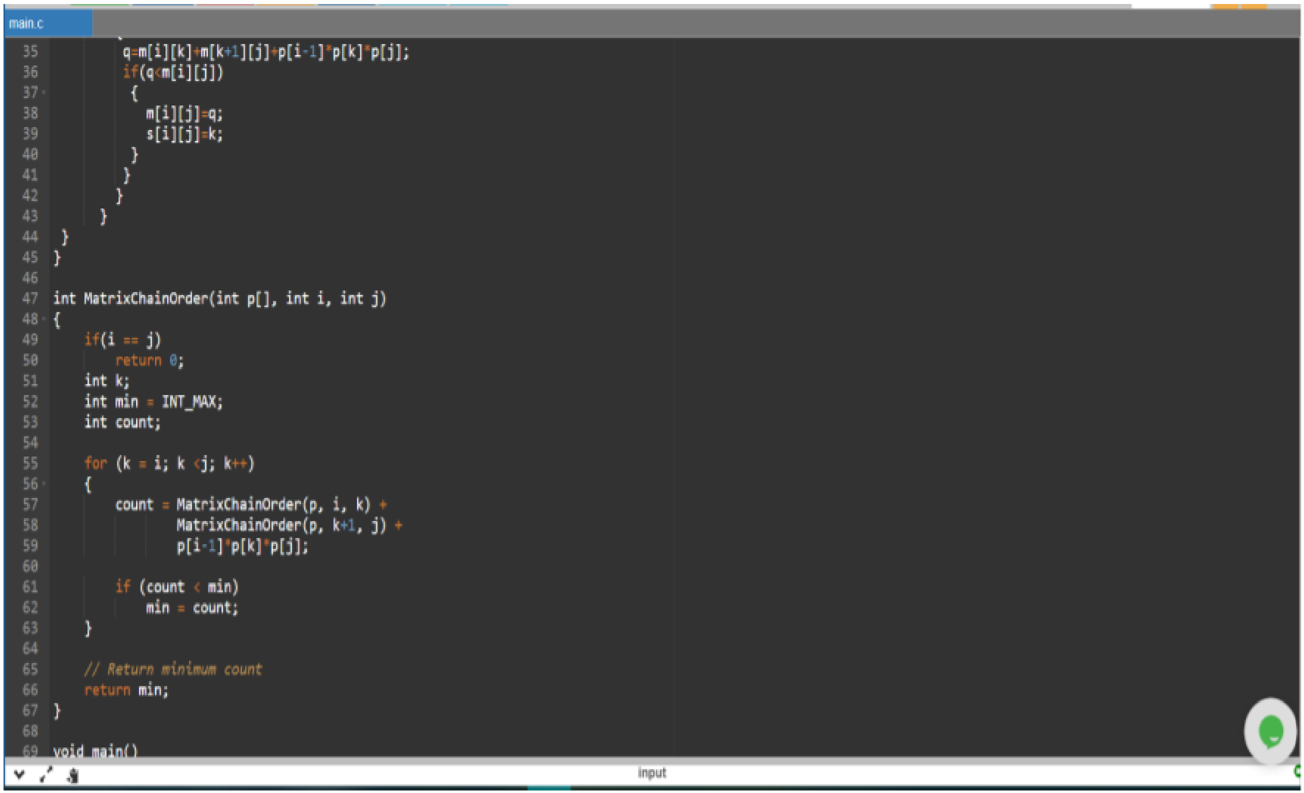
SUBJECT CODE-18CSC204J

SUBMITTED TO- Mr. Himansu Sekhar Pattnayak

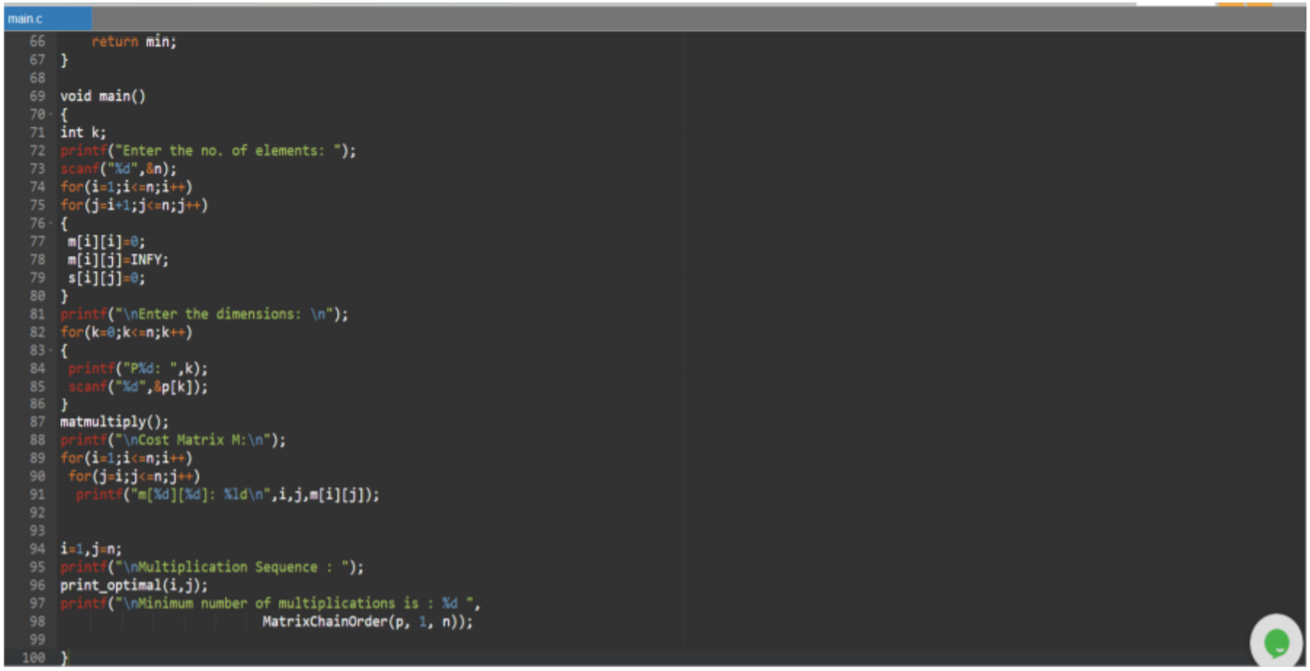
Implement the matrix chain multiplication program using dynamic programming.
The dimensions of the matrices are P=<30,35,15,5,10,20,25>.



```c
#include <stdio.h>
#include<limits.h>
#define INFY 999999999
long int m[20][20];
int s[20][20];
int p[20],i,j,n;

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

void matmultiply(void)
{
long int q;
int k;
for(i=n;i>0;i--)
 {
   for(j=i;j<=n;j++)
    {
     if(i==j)
       m[i][j]=0;
     else
       {
        for(k=i;k<j;k++)
        {
         q=m[i][k]+m[k+1][j]+p[i-1]*p[k]*p[j];
```

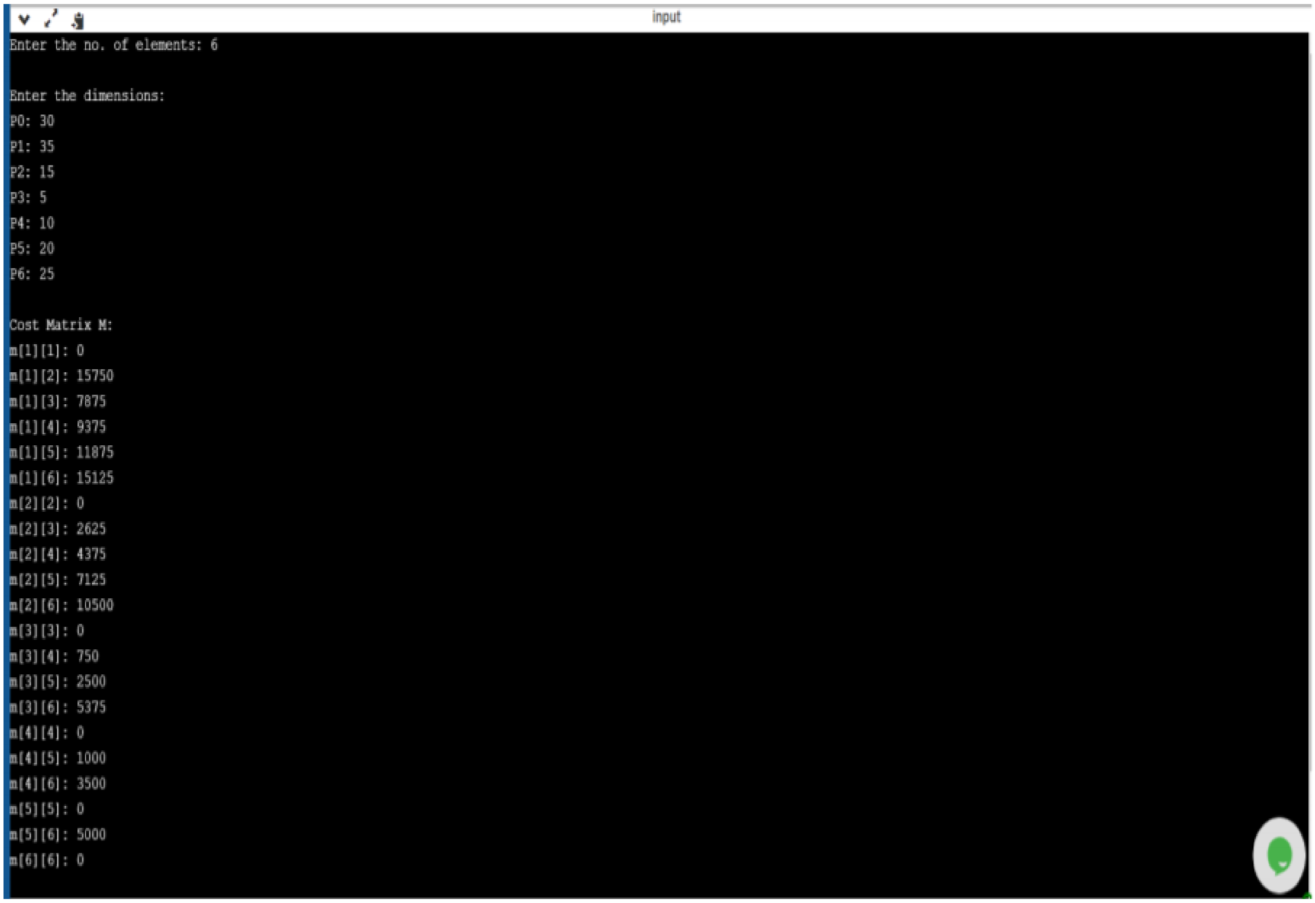


```
         q=m[i][k]+m[k+1][j]+p[i-1]*p[k]*p[j];
         if(q<m[i][j])
          {
            m[i][j]=q;
            s[i][j]=k;
          }
         }
        }
      }
 }
}

int MatrixChainOrder(int p[], int i, int j)
{
    if(i == j)
        return 0;
    int k;
    int min = INT_MAX;
    int count;

    for (k = i; k <j; k++)
    {
        count = MatrixChainOrder(p, i, k) +
                MatrixChainOrder(p, k+1, j) +
                p[i-1]*p[k]*p[j];

        if (count < min)
            min = count;
    }

    // Return minimum count
    return min;
}

void main()
```

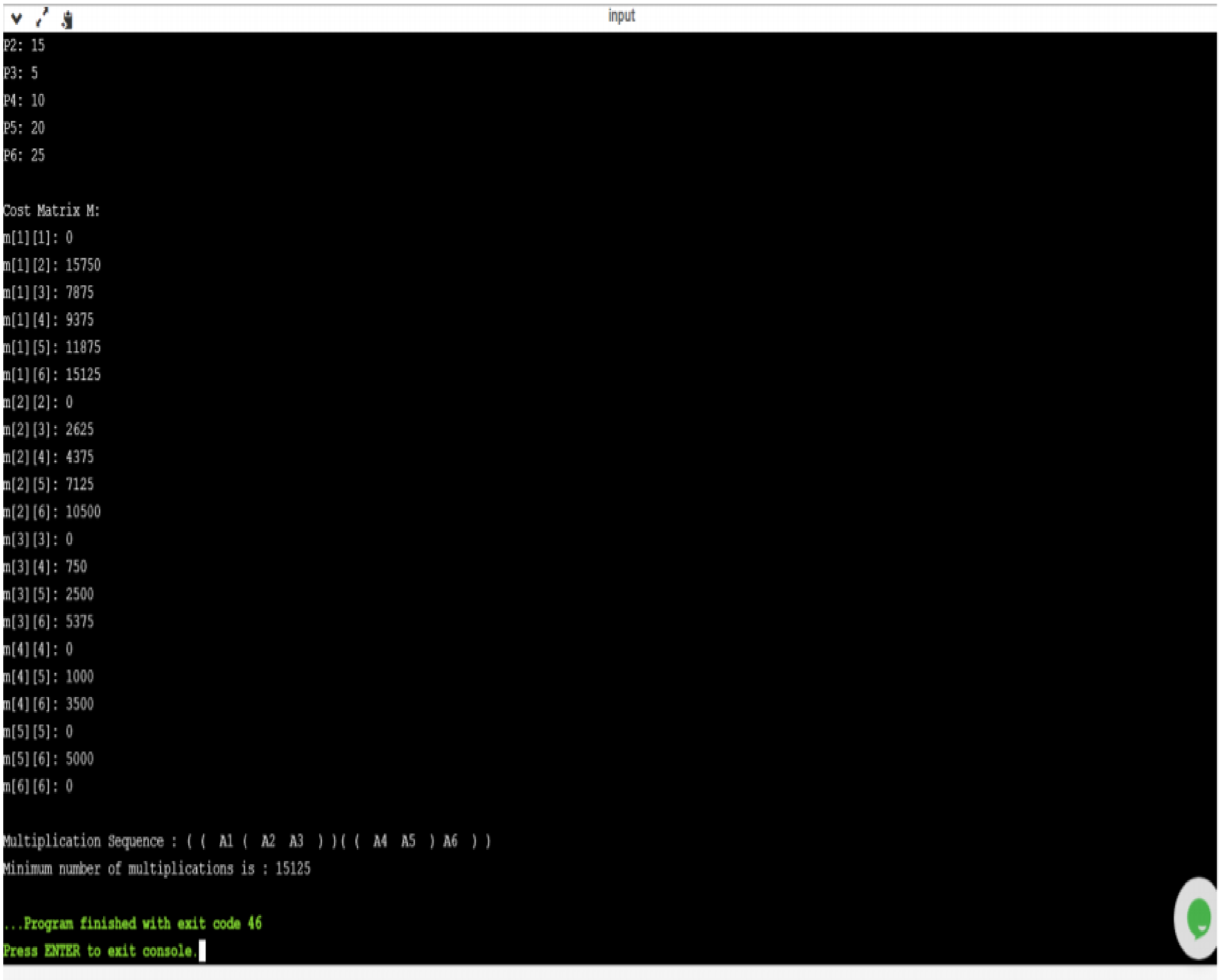


```
    return min;
}

void main()
{
int k;
printf("Enter the no. of elements: ");
scanf("%d",&n);
for(i=1;i<=n;i++)
for(j=i+1;j<=n;j++)
{
 m[i][i]=0;
 m[i][j]=INFY;
 s[i][j]=0;
}
printf("\nEnter the dimensions: \n");
for(k=0;k<=n;k++)
{
 printf("P%d: ",k);
 scanf("%d",&p[k]);
}
matmultiply();
printf("\nCost Matrix M:\n");
for(i=1;i<=n;i++)
 for(j=i;j<=n;j++)
  printf("m[%d][%d]: %ld\n",i,j,m[i][j]);


i=1,j=n;
printf("\nMultiplication Sequence : ");
print_optimal(i,j);
printf("\nMinimum number of multiplications is : %d ",
                          MatrixChainOrder(p, 1, n));

}
```

OUTPUT: -



```
Enter the no. of elements: 6

Enter the dimensions:
P0: 30
P1: 35
P2: 15
P3: 5
P4: 10
P5: 20
P6: 25

Cost Matrix M:
m[1][1]: 0
m[1][2]: 15750
m[1][3]: 7875
m[1][4]: 9375
m[1][5]: 11875
m[1][6]: 15125
m[2][2]: 0
m[2][3]: 2625
m[2][4]: 4375
m[2][5]: 7125
m[2][6]: 10500
m[3][3]: 0
m[3][4]: 750
m[3][5]: 2500
m[3][6]: 5375
m[4][4]: 0
m[4][5]: 1000
m[4][6]: 3500
m[5][5]: 0
m[5][6]: 5000
m[6][6]: 0
```



```
P2: 15
P3: 5
P4: 10
P5: 20
P6: 25

Cost Matrix M:
m[1][1]: 0
m[1][2]: 15750
m[1][3]: 7875
m[1][4]: 9375
m[1][5]: 11875
m[1][6]: 15125
m[2][2]: 0
m[2][3]: 2625
m[2][4]: 4375
m[2][5]: 7125
m[2][6]: 10500
m[3][3]: 0
m[3][4]: 750
m[3][5]: 2500
m[3][6]: 5375
m[4][4]: 0
m[4][5]: 1000
m[4][6]: 3500
m[5][5]: 0
m[5][6]: 5000
m[6][6]: 0

Multiplication Sequence : ( (  A1 (  A2  A3  ) )( (  A4  A5  ) A6  ) )
Minimum number of multiplications is : 15125

...Program finished with exit code 46
Press ENTER to exit console.
```