**PROGRAM-14**

**AIM –Write an algorithm and program to implement Matrix-Chain Multiplication.**

**ALGORITHM-**

Matrix Multiply(A, B):

Assume dimension of A is (m x n), dimension of B is (p x q)

Begin

   if n is not same as p, then exit

   otherwise define C matrix as (m x q)

   for i in range 0 to m - 1, do

      for j in range 0 to q – 1, do

         for k in range 0 to p, do

            C[i, j] = C[i, j] + (A[i, k] \* A[k, j])

         done

      done

   done

End

**SOURCE CODE-**

#include<stdio.h>

#include<conio.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];

if(q<m[i][j])

{

m[i][j]=q;

s[i][j]=k;

}

}

}

}

}

}

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]);

printf("\nMatrix S for k values:\n");

for(i=1;i<=n;i++)

for(j=i;j<=n;j++)

printf("m[%d][%d]: %d\n",i,j,s[i][j]);

i=1,j=n;

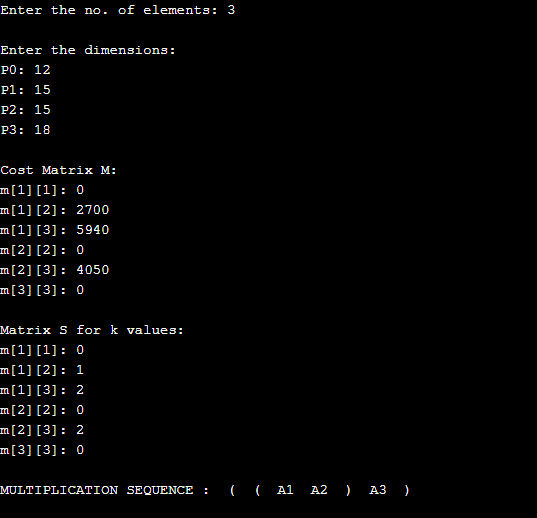
printf("\nMULTIPLICATION SEQUENCE : ");

print\_optimal(i,j);

getch();

}

**OUTPUT-**

****