**Program No.:-5**

**AIM: Write a program to solve the problem of Matrix chain multiplication.**

**Source Code:**

#include<stdio.h>

#include<conio.h>

#include<limits.h>

int p[10],n,s[10][10];

long m[10][10];

void matrix\_chain\_order()

{

int i,j=0,k;

long q=0;

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

m[i][i]=0;

for(int l=2;l<=n;l++)

{

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

{

j=i+l-1;

m[i][j]=LONG\_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;

s[i][j]=k;

}

}

}

}

}

void print\_optimal\_parens(int i,int j)

{

if(i==j)

printf("A%d",i);

else

{

printf("(");

print\_optimal\_parens(i,s[i][j]);

print\_optimal\_parens(s[i][j]+1,j);

printf(")");

}

}

int main()

{

clrscr();

int a,b,j,k;

printf("enter no.of arrays");

scanf("%d",&n);

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

{

printf("enter size of matrix");

scanf("%d%d",&a,&b);

p[i-1]=a;

p[i]=b;

}

matrix\_chain\_order();

printf<<"table of m[i][j]"<<endl;

for(j=n;j>=1;j--)

{

for(k=1;k<=j;k++)

printf("%d\t",m[k][j]);

printf("\n");

}

printf<<"min number of multiplications:%d\n",m[1][n]);

printf("table of s[i][j]\n");

for(j=n;j>=1;j--)

{

for(k=1;k<j;k++)

printf("%d\t",s[k][j]);

printf("\n");;

}

printf("\n optimal multiplication order:");

print\_optimal\_parens(1,n);

getch();

}

**Output:**

