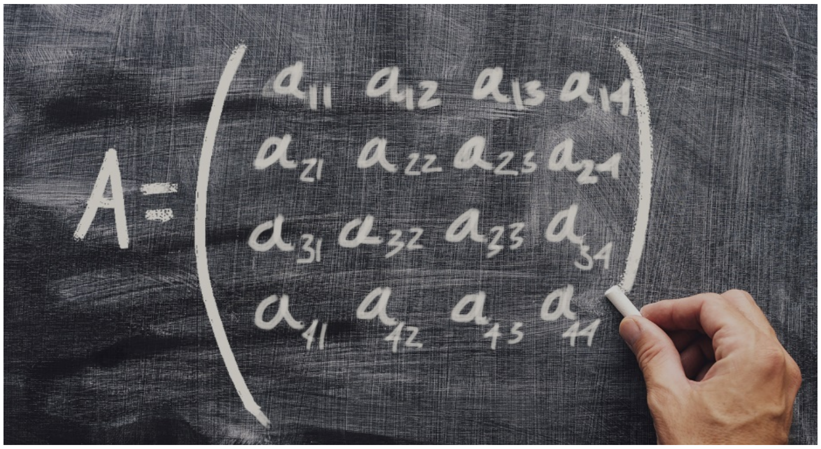
19 de noviembre del 2021

Práctica 2

Producto de matrices en cadena



MIGUEL ANGEL BALTAZAR COYOTL

ESCUELA SUPERIOR DE CÓMPUTO

Introducción

Para poder multiplicar matrices además de el trabajo que se debe hacer para multiplicar cada matriz también importa el orden en la que se multiplican las matrices, esto es muy importante ya que al tener un orden en especifico se puede ahorrar mucho trabajo lo cual se ve en programación dinámica, por este motivo es que se diseño este programa en el cual se muestra la forma en la que es más efectiva multiplicar las matrices, esto se hace tomando el que genere el menor número de pasos que se deben realizar.

Código en C++

```
#include<iostream>

using namespace std;

struct bases{

        int n1;

        int n2;

}base[100];

int mult(bases, bases);

bases nueva(bases,bases,bases);

void tresmat();

void fourmat();

void fivemat();

void res(bases m1);

int min(int,int);

void color(int,int);

int i=0;
int main(){
    int matrices;
    system("CLS");
    cout<<"Ingrese el numero de matrices: ";
    cin>>matrices;

    for(i=0;i<matrices;i++){
        cout<<"Ingrese el orden de su matriz: ";
        cin>>base[i].n1>>base[i].n2;
    }
    switch(matrices){
        case 3:
            tresmat();
            break;
        case 4:
            fourmat();
            break;
        case 5:
            fivemat();
            break;
    }
    return 0;
}

bases nueva(bases m1,bases m2){
    bases j;
    j.n1 = m1.n1;
    j.n2 = m2.n2;
    return j;
}
```

```
int mult(bases m1,bases m2){
        int mu = m1.n1 * m1.n2;
        return mu * m2.n2;
}
void res(bases m1){
        cout<<m1.n1<<"x"<<m1.n2;
        cout<<" ";
}
int min(int a,int b){
        if(a < b) return a;
        else return b;
}
void color(int a,int b){
        if(a == b){
                printf("\033[0;31m");
        }
}

void tresmat(){
        // Producto de matrices en cadena
        int b1,c2,c1a,c1b;
        b1 = mult(base[0],base[1]);
        base[i] = nueva(base[0],base[1]);
        c1a = mult(base[i],base[2]) + b1;
        c2 = mult(base[1],base[2]);
        i++;
        base[i] = nueva(base[1],base[2]);
        c1b = mult(base[0],base[i]) + c2;

        if(min((c1a),(c1b)) == (c1a) ){
                printf("\033[0;31m");
                cout<<endl<<"(AB)"<<"C"<<"-> "<<c1a;printf("\033[0m");
        }
        else{
                printf("\033[0;31m");
                cout<<"\n\tA"<<"(BC)"<<"-> "<<c1b;printf("\033[0m");
        }

        // Creacion de la tabla
        cout<<"\n___________________________________________________________________________"<<endl;
        cout<<"   |  \t A"<<"\t         B"<<"\t              C";
        cout<<"\n----------------------------------------------------------------";
        cout<<"\nA  |     X      ";
        cout<<"A";res(base[0]);cout<<"B";res(base[1]);cout<<"-> "<<b1;
        color(min((c1a),(c1b)),c1a);
        cout<<"\t(AB)";res(base[3]);cout<<"C";res(base[2]);cout<<"-> "<<c1a;printf("\033[0m");
        cout<<endl<<"   |";
        color(min((c1a),(c1b)),c1b);
        cout<<"\t\t\t\t\tA";res(base[0]);cout<<"(BC)";res(base[4]);cout<<"-> "<<c1b;printf("\033[0m");
        cout<<"\n----------------------------------------------------------------";
        cout<<"\nB  |     X              X       ";
        cout<<"\tB";res(base[1]);cout<<"C";res(base[2]);cout<<"-> "<<c2;
        cout<<"\n----------------------------------------------------------------";
        cout<<"\nC  |     X              X     \t   X       "<<endl;
        cout<<"___________________________________________________________________________"<<endl;
}

void fourmat(){
        // Producto de matrices en cadena
        int b1,c2,c1a,c1b,j,d3,k,d2b,d2a,d1a,d1b,c1,d2;
        b1 = mult(base[0],base[1]);
        //4
```

```
base[i] = nueva(base[0],base[1]);
c1a = mult(base[i],base[2]) + b1;
i++; //5
base[i] = nueva(base[4],base[2]);
c2 = mult(base[1],base[2]);
i++;//6
base[i] = nueva(base[1],base[2]);
c1b = mult(base[0],base[i]) + c2;
i++;//7
base[i] = nueva(base[0],base[6]);
if(min((c1a),(c1b)) == (c1a) ){
        j = 5;
        c1 = c1a;
}
else{
        j = 7;
        c1 = c1b;
}
d3 = mult(base[2],base[3]);
i++; //8
base[i] = nueva(base[2],base[3]);
d2b = mult(base[1],base[i]) + d3;
i++; //9
base[i] = nueva(base[1],base[8]);
d2a = mult(base[6],base[3]) + c2;
i++; //10
base[i] = nueva(base[6],base[3]);

if(min((d2a),(d2b)) == (d2a) ){
        k = 10;
        d2 = d2a;
```

```
    }
    else{
        k = 9;
        d2 = d2b;
    }
    d1a = mult(base[j],base[3]) + c1;
    i++; //11
    base[i] = nueva(base[j],base[3]);
    d1b = mult(base[0],base[k]) + d2;
    i++;//12
    base[i] = nueva(base[0],base[k]);
    if(min((d1a),(d1b)) == (d1a) ){
        printf("\033[0;31m");
        cout<<endl<<"(A(BC))"<<"D"<<"-> "<<d1a;printf("\033[0m");
    }
    else{
        printf("\033[0;31m");
        cout<<"\n\tA"<<"((BC)D)"<<"-> "<<d1b;printf("\033[0m");
    }
    // Creacion de la tabla
    cout<<"\n________________________________________________________________________________
______________________________"<<endl;
    cout<<"   |  \t A"<<"\t        B"<<"\t            C"<<"\t                    D";
    cout<<"\n---------------------------------------------------------------------------------------------------------";
    cout<<"\nA  |     X       ";
    cout<<"A";res(base[0]);cout<<"B";res(base[1]);cout<<"-> "<<b1;
    color(min((c1a),(c1b)),c1a);
    cout<<"\t(AB)";res(base[4]);cout<<"C";res(base[2]);cout<<"-> "<<c1a;printf("\033[0m");
    color(min(d1a,d1b),d1a);
    cout<<"\t(A(BC))";res(base[j]);cout<<" D";res(base[3]);cout<<"-> "<<d1a;printf("\033[0m");
    cout<<endl<<"   |";
```

```
	color(min(c1b,c1a),c1b);

	cout<<"\t\t\t\t\tA";res(base[0]);cout<<"(BC)";res(base[6]);cout<<"-> "<<c1b;printf("\033[0m");

	color(min(d1a,d1b),d1b);

	cout<<"\t\tA";res(base[0]);cout<<"((BC)D)";res(base[k]);cout<<"-> "<<d1b;printf("\033[0m");

	cout<<"\n-------------------------------------------------------------------------------------------------";

	cout<<"\nB  |    X              X        ";

	cout<<"\tB";res(base[1]);cout<<"C";res(base[2]);cout<<"-> "<<c2;

	color(min(d2a,d2b),d2a);

	cout<<"\t\t(BC)";res(base[6]);cout<<"D";res(base[3]);cout<<"-> "<<d2a;printf("\033[0m");

	cout<<endl<<"   |";

	color(min(d2a,d2b),d2b);

	cout<<"\t\t\t\t\t\t\t\t\tB";res(base[1]);cout<<"(CD)";res(base[8]);cout<<"->
"<<d2b;printf("\033[0m");

	cout<<"\n-------------------------------------------------------------------------------------------------";

	cout<<"\nC  |    X              X     \t\t   X        ";

	cout<<"\t\tC";res(base[2]);cout<<"D";res(base[3]);cout<<"-> "<<d3<<endl;

	cout<<"-------------------------------------------------------------------------------------------------";

	cout<<"\nD  |    X              X     \t\t   X     \t\t\t   X";

	cout<<"\n________________________________________________________________________________
______________________________"<<endl;

}


void fivemat(){

	// Producto de matrices en cadena

	int b1,c2,c1a,c1b,j,d3,k,d2b,d2a,d1a,d1b,e4,e3b,e3a,e2b,e2a,e1b,l,m,n,e1a,c1,d2,d1,e3,e2;

	b1 = mult(base[0],base[1]);

	//5

	base[i] = nueva(base[0],base[1]);

	c1a = mult(base[i],base[2]) + b1;

	i++; //6

	base[i] = nueva(base[5],base[2]);
```

```
c2 = mult(base[1],base[2]);

i++;//7

base[i] = nueva(base[1],base[2]);

c1b = mult(base[0],base[i]) + c2;

i++;//8

base[i] = nueva(base[0],base[7]);

if(min(c1a,c1b) == c1a){

        j = 6;

        c1 = c1a;

}

else{

        j = 8;

        c1 = c1b;

}

d3 = mult(base[2],base[3]);

i++; //9

base[i] = nueva(base[2],base[3]);

d2b = mult(base[1],base[i]) + d3;

i++; //10

base[i] = nueva(base[1],base[9]);

d2a = mult(base[7],base[3]) + c2;

i++; //11

base[i] = nueva(base[7],base[3]);

if(min(d2a,d2b) == d2a ){

        k = 11;

        d2 = d2a;

}

else{

        k = 10;

        d2 = d2b;

}
```

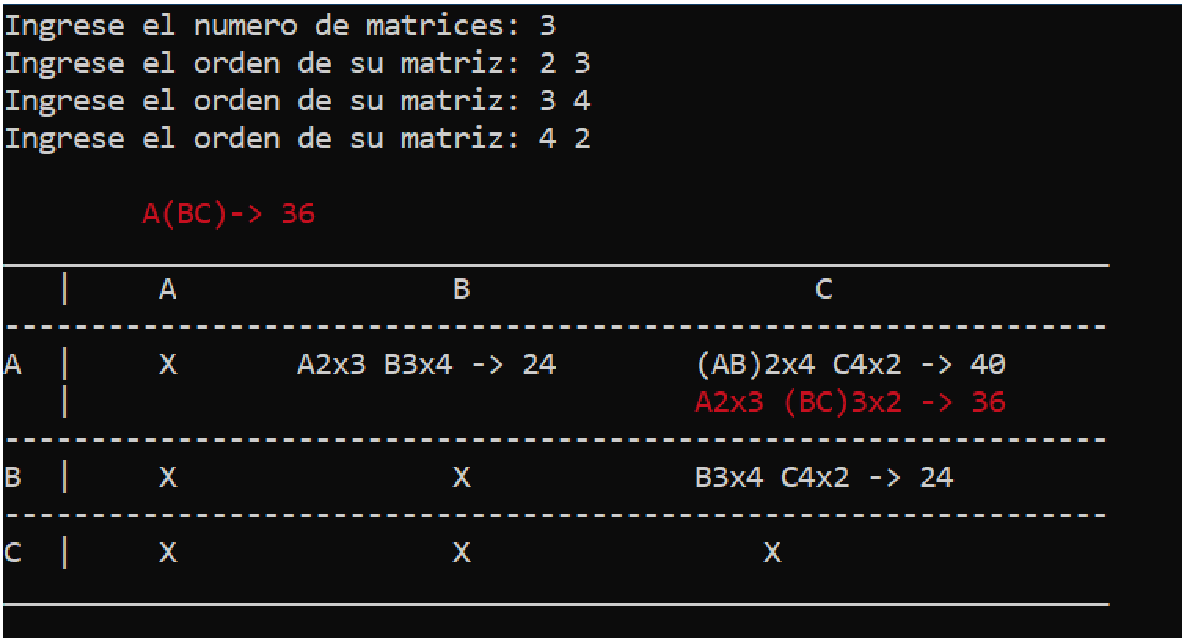
```
d1a = mult(base[j],base[3]) + c1;
i++; //12
base[i] = nueva(base[j],base[3]);
d1b = mult(base[0],base[k]) + d2;
i++;//13
base[i] = nueva(base[0],base[k]);
if(min(d1a,d1b) == d1a ){
        l = 12;
        d1 = d1a;
}
else{
        l = 13;
        d1 = d1b;
}
e4 = mult(base[3],base[4]);
i++;//14
base[i] = nueva(base[3],base[4]);
e3b = mult(base[2],base[i]) + e4;
i++;//15
base[i] = nueva(base[2],base[14]);
e3a = mult(base[9],base[4]) + d3;
i++;//16
base[i] = nueva(base[9],base[4]);
if(min(e3a,e3b) == e3a ){
        m = 16;
        e3 = e3a;
}
else{
        m = 15;
        e3 = e3b;
}
```

```
    e2b = mult(base[1],base[m]) + e3;

    i++;//17

    base[i] = nueva(base[1],base[m]);

    e2a = mult(base[k],base[4]) + d2;

    i++;//18

    base[i] = nueva(base[k],base[4]);

    if(min(e2a,e2b) == e2a){

        n = 18;

        e2 = e2a;

    }

    else{

        n = 17;

        e2 = e2b;

    }

    e1b = mult(base[0],base[n]) + e2;

    e1a = mult(base[l],base[4]) + d1;

    if(min(e1a,e1b) == e1a){

        printf("\033[0;31m");

        cout<<endl<<"((A(BC))D)"<<"E"<<"-> "<<e1a;printf("\033[0m");

    }

    else{

        printf("\033[0;31m");

        cout<<"\n\tA"<<"(((BC)D)E)"<<"-> "<<e1b;printf("\033[0m");

    }

    // Creacion de la tabla

    cout<<"\n______________________________________________________________________
__________________________________________________________"<<endl;

    cout<<"   |  \t A"<<"\t        B"<<"\t            C"<<"\t                  D"<<"\t                   E";

    cout<<"\n-----------------------------------------------------------------------------------------------------------------------------
-------------";

    cout<<"\nA  |     X       ";
```

```
        cout<<"A";res(base[0]);cout<<"B";res(base[1]);cout<<"-> "<<b1;

        color(min(c1b,c1a),c1a);

        cout<<"\t(AB)";res(base[5]);cout<<"C";res(base[2]);cout<<"-> "<<c1a;printf("\033[0m");

        color(min(d1a,d1b),d1a);

        cout<<"\t\t(A(BC))";res(base[j]);cout<<" D";res(base[3]);cout<<"-> "<<d1a;printf("\033[0m");

        color(min(e1a,e1b),e1a);

        cout<<"\t\t((A(BC))D)";res(base[l]);cout<<" E";res(base[4]);cout<<"-> "<<e1a;printf("\033[0m");

        cout<<endl<<"   |";

        color(min(c1b,c1a),c1b);

        cout<<"\t\t\t\t\tA";res(base[0]);cout<<"(BC)";res(base[7]);cout<<"-> "<<c1b;printf("\033[0m");

        color(min(d1a,d1b),d1b);

        cout<<"\t\tA";res(base[0]);cout<<"((BC)D)";res(base[k]);cout<<"-> "<<d1b;printf("\033[0m");

        color(min(e1a,e1b),e1b);

        cout<<"\t\tA";res(base[0]);cout<<"(((BC)D)E)";res(base[n]);cout<<"-> "<<e1b;printf("\033[0m");

        cout<<"\n--------------------------------------------------------------------------------------------------------------
-------------";

        cout<<"\nB  |     X               X        ";

        cout<<"\tB";res(base[1]);cout<<"C";res(base[2]);cout<<"-> "<<c2;

        color(min(d2a,d2b),d2a);

        cout<<"\t\t\t(BC)";res(base[7]);cout<<"D";res(base[3]);cout<<"-> "<<d2a;printf("\033[0m");

        color(min(e2a,e2b),e2a);

        cout<<"\t\t((BC)D)";res(base[k]);cout<<"E";res(base[4]);cout<<"-> "<<e2a;printf("\033[0m");

        cout<<endl<<"   |";

        color(min(d2a,d2b),d2b);

        cout<<"\t\t\t\t\t\t\t\t\tB";res(base[1]);cout<<"(CD)";res(base[9]);cout<<"->
"<<d2b;printf("\033[0m");

        color(min(e2a,e2b),e2b);

        cout<<"\t\tB";res(base[1]);cout<<"((CD)E)";res(base[m]);cout<<"-> "<<e2b;printf("\033[0m");

        cout<<"\n--------------------------------------------------------------------------------------------------------------
-------------";

        cout<<"\nC  |     X               X      \t\t    X        ";

        cout<<"\t\tC";res(base[2]);cout<<"D";res(base[3]);cout<<"-> "<<d3;
        color(min(e3a,e3b),e3a);

        cout<<"\t\t\t(CD)";res(base[9]);cout<<"E";res(base[4]);cout<<"-> "<<e3a;printf("\033[0m");

        cout<<endl<<"   |";

        color(min(e3a,e3b),e3b);

        cout<<"\t\t\t\t\t\t\t\t\t\t\t\t\tC";res(base[2]);cout<<"(DE)";res(base[14]);cout<<"-> "<<e3b;printf("\033[0m");

        cout<<"\n------------------------------------------------------------------------------------------------------------------------";

        cout<<"\nD  |    X             X    \t\t   X    \t\t\t   X";

        cout<<"\t\t\tD";res(base[3]);cout<<"E";res(base[4]);cout<<"-> "<<e4;

        cout<<"\n------------------------------------------------------------------------------------------------------------------------";

        cout<<"\nE  |    X             X    \t\t   X    \t\t\t   X    \t\t   X";

        cout<<"\n____________________________________________________________________________________________________________________________________________________________________________"<<endl;

}
```

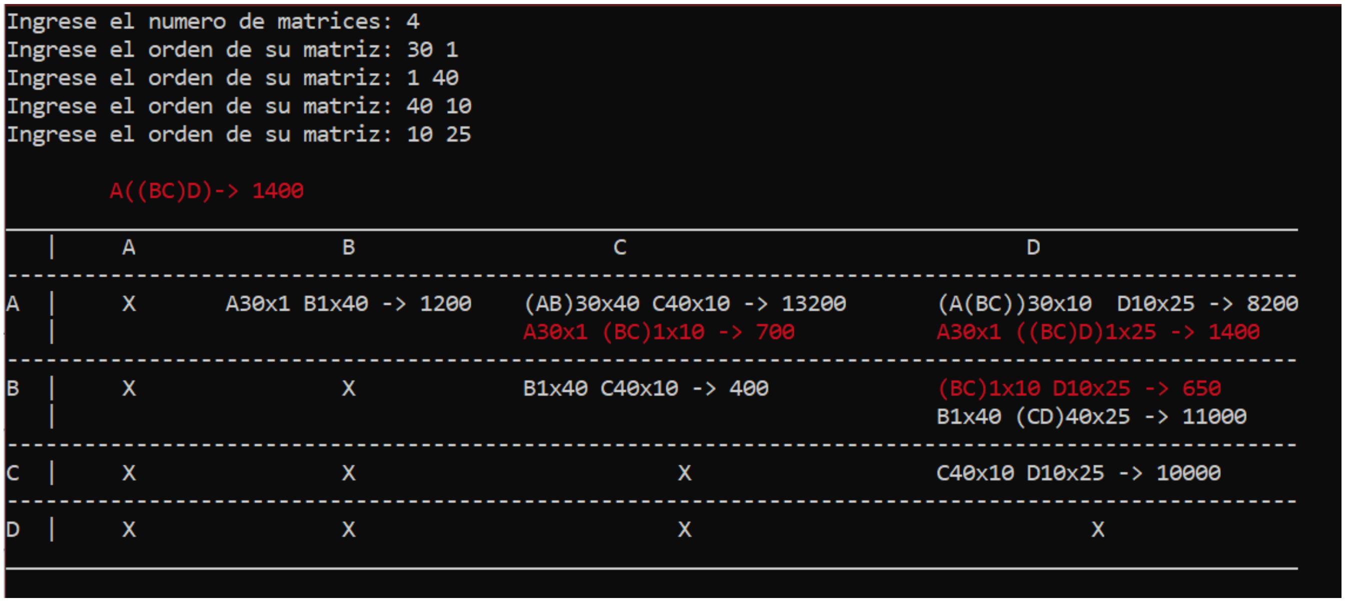
Pruebas especificadas

Prueba 1:



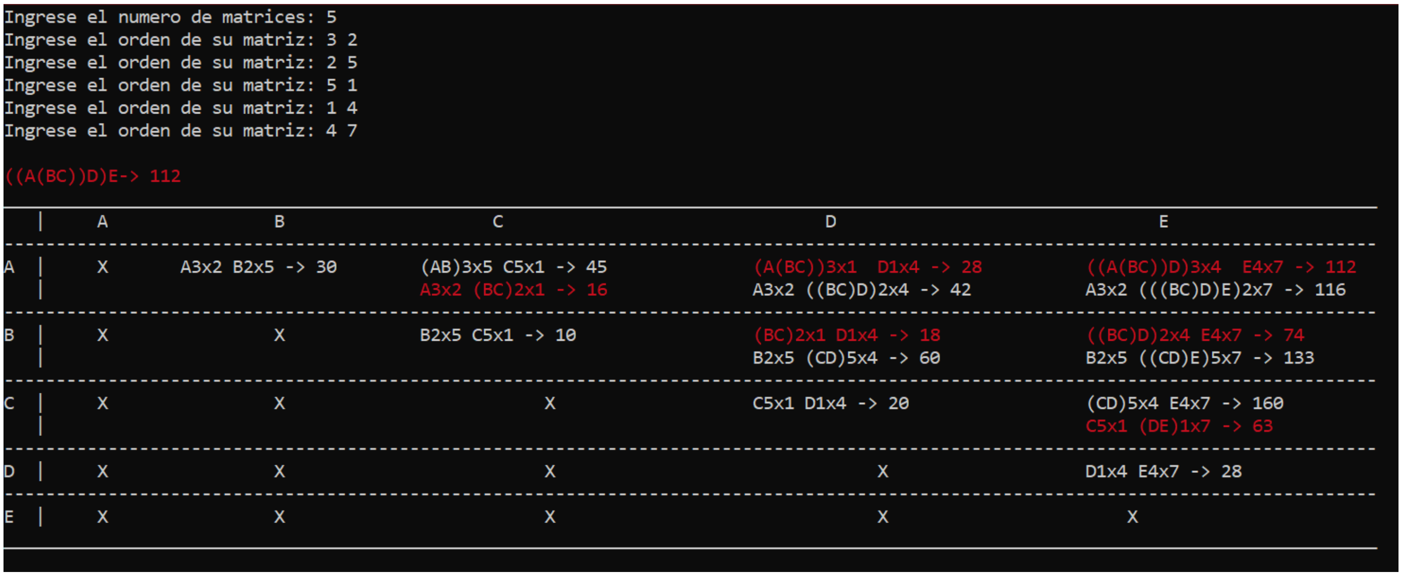
```
Ingrese el numero de matrices: 3
Ingrese el orden de su matriz: 2 3
Ingrese el orden de su matriz: 3 4
Ingrese el orden de su matriz: 4 2

        A(BC)-> 36

────────────────────────────────────────────────────────────────
   |     A                B                    C
----------------------------------------------------------------
A  |     X       A2x3 B3x4 -> 24        (AB)2x4 C4x2 -> 40
   |                                    A2x3 (BC)3x2 -> 36
----------------------------------------------------------------
B  |     X                X             B3x4 C4x2 -> 24
----------------------------------------------------------------
C  |     X                X                 X

────────────────────────────────────────────────────────────────
```

Prueba 2:



```
Ingrese el numero de matrices: 4
Ingrese el orden de su matriz: 30 1
Ingrese el orden de su matriz: 1 40
Ingrese el orden de su matriz: 40 10
Ingrese el orden de su matriz: 10 25

        A((BC)D)-> 1400

────────────────────────────────────────────────────────────────────────────────────────────────────
   |     A                B                    C                               D
----------------------------------------------------------------------------------------------------
A  |     X       A30x1 B1x40 -> 1200    (AB)30x40 C40x10 -> 13200       (A(BC))30x10  D10x25 -> 8200
   |                                    A30x1 (BC)1x10 -> 700           A30x1 ((BC)D)1x25 -> 1400
----------------------------------------------------------------------------------------------------
B  |     X                X             B1x40 C40x10 -> 400             (BC)1x10 D10x25 -> 650
   |                                                                    B1x40 (CD)40x25 -> 11000
----------------------------------------------------------------------------------------------------
C  |     X                X                         X                   C40x10 D10x25 -> 10000
----------------------------------------------------------------------------------------------------
D  |     X                X                         X                               X

────────────────────────────────────────────────────────────────────────────────────────────────────
```

Prueba 3:



```
Ingrese el numero de matrices: 5
Ingrese el orden de su matriz: 3 2
Ingrese el orden de su matriz: 2 5
Ingrese el orden de su matriz: 5 1
Ingrese el orden de su matriz: 1 4
Ingrese el orden de su matriz: 4 7

((A(BC))D)E-> 112
────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
   |     A                B                     C                               D                               E
------------------------------------------------------------------------------------------------------------------------------------
A  |     X       A3x2 B2x5 -> 30        (AB)3x5 C5x1 -> 45              (A(BC))3x1  D1x4 -> 28          ((A(BC))D)3x4  E4x7 -> 112
   |                                    A3x2 (BC)2x1 -> 16              A3x2 ((BC)D)2x4 -> 42           A3x2 (((BC)D)E)2x7 -> 116
------------------------------------------------------------------------------------------------------------------------------------
B  |     X                X             B2x5 C5x1 -> 10                 (BC)2x1 D1x4 -> 18              ((BC)D)2x4 E4x7 -> 74
   |                                                                    B2x5 (CD)5x4 -> 60              B2x5 ((CD)E)5x7 -> 133
------------------------------------------------------------------------------------------------------------------------------------
C  |     X                X                          X                  C5x1 D1x4 -> 20                 (CD)5x4 E4x7 -> 160
   |                                                                                                    C5x1 (DE)1x7 -> 63
------------------------------------------------------------------------------------------------------------------------------------
D  |     X                X                          X                               X                  D1x4 E4x7 -> 28
------------------------------------------------------------------------------------------------------------------------------------
E  |     X                X                          X                               X                       X
────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
```