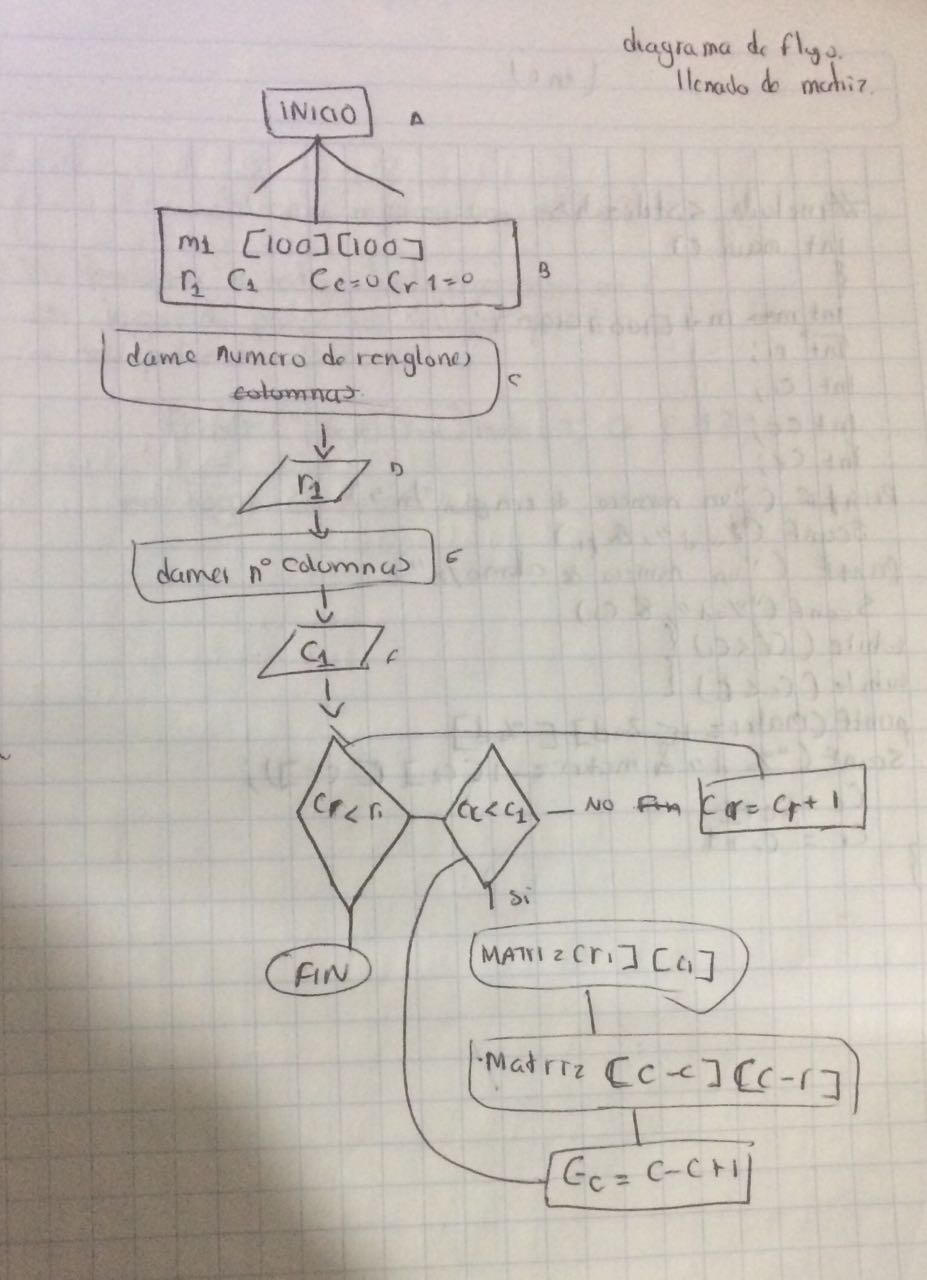
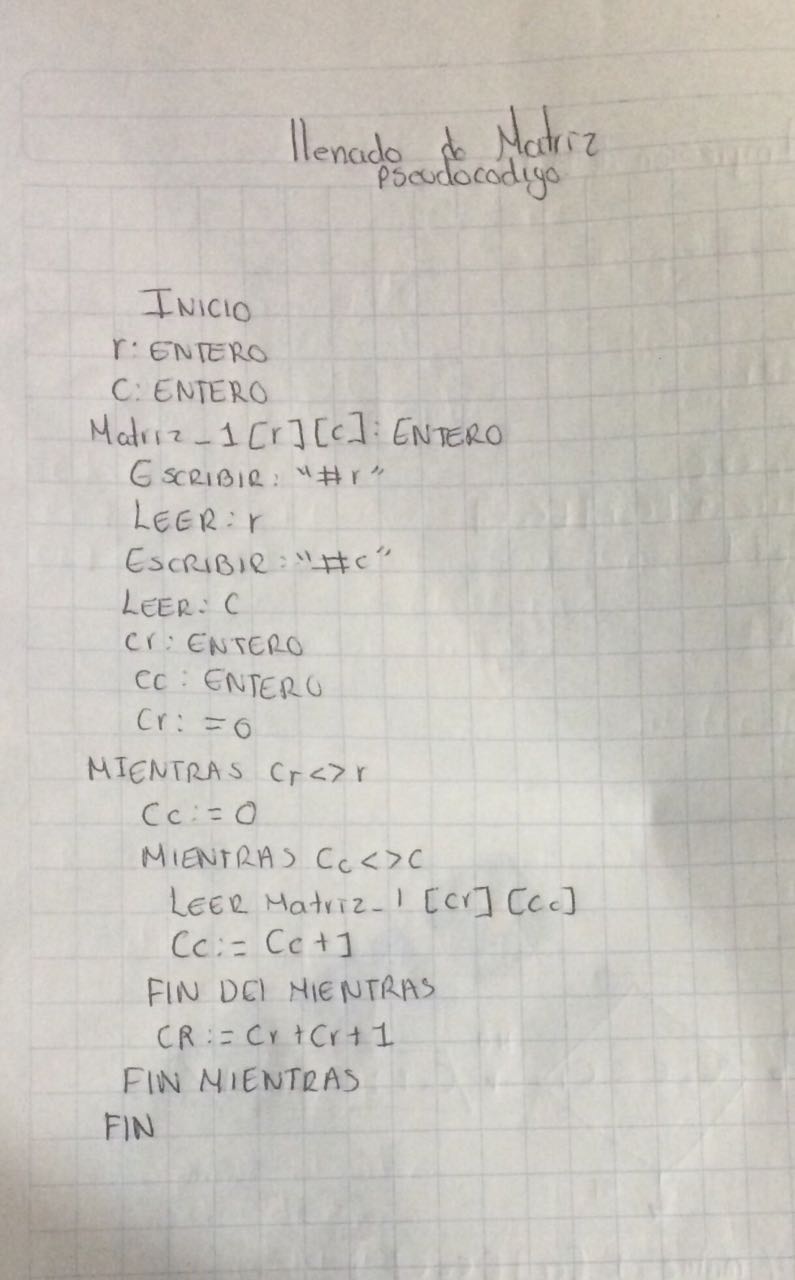
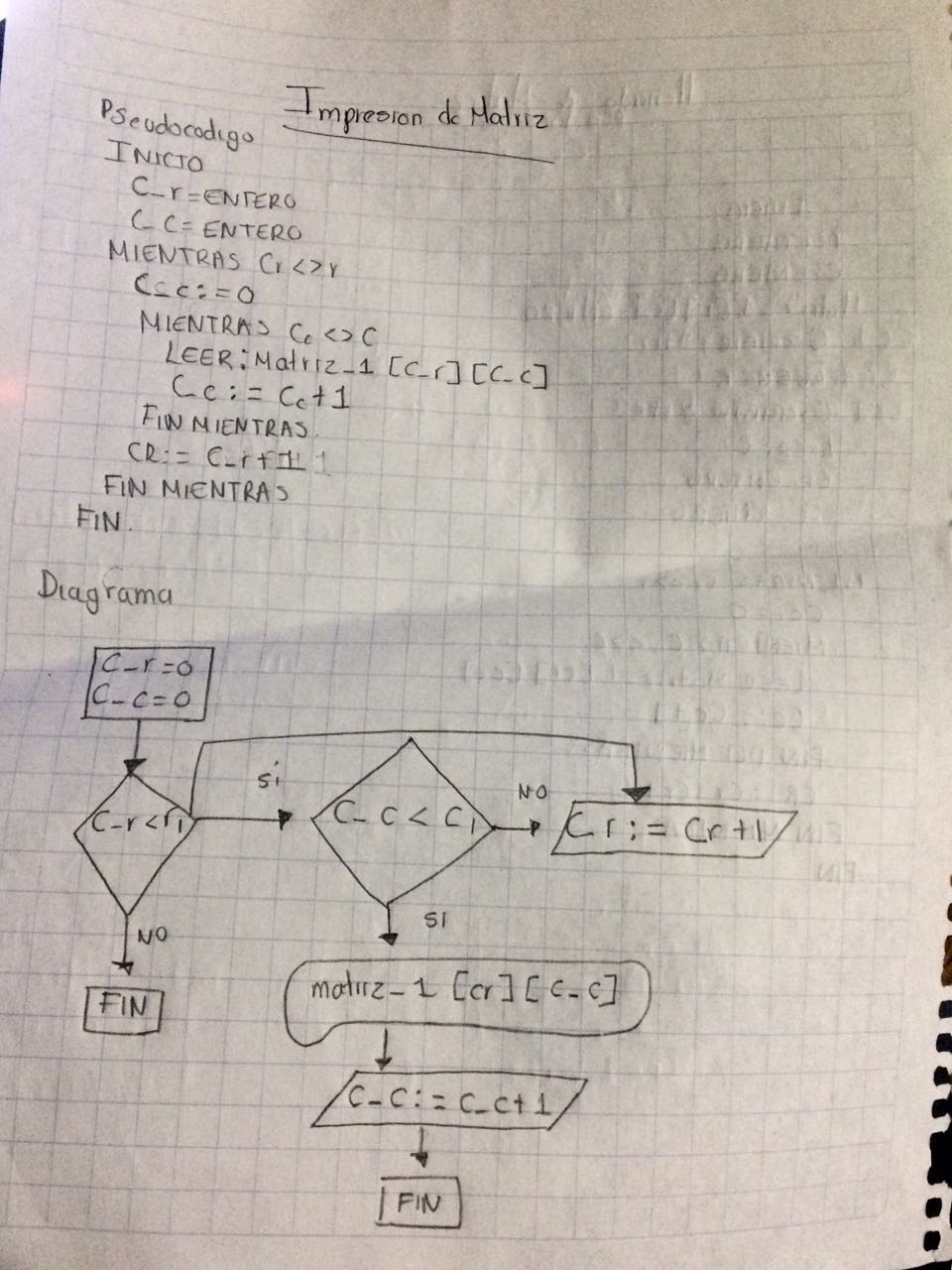
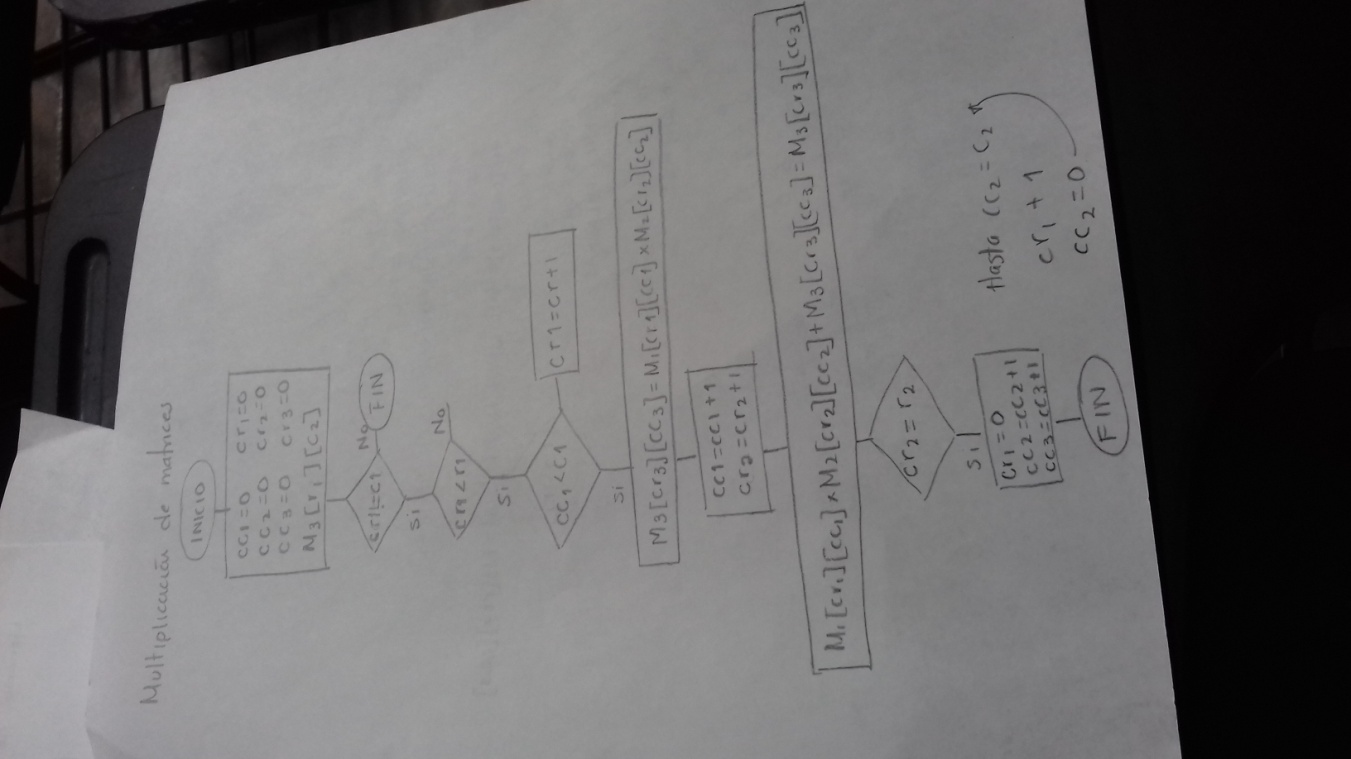
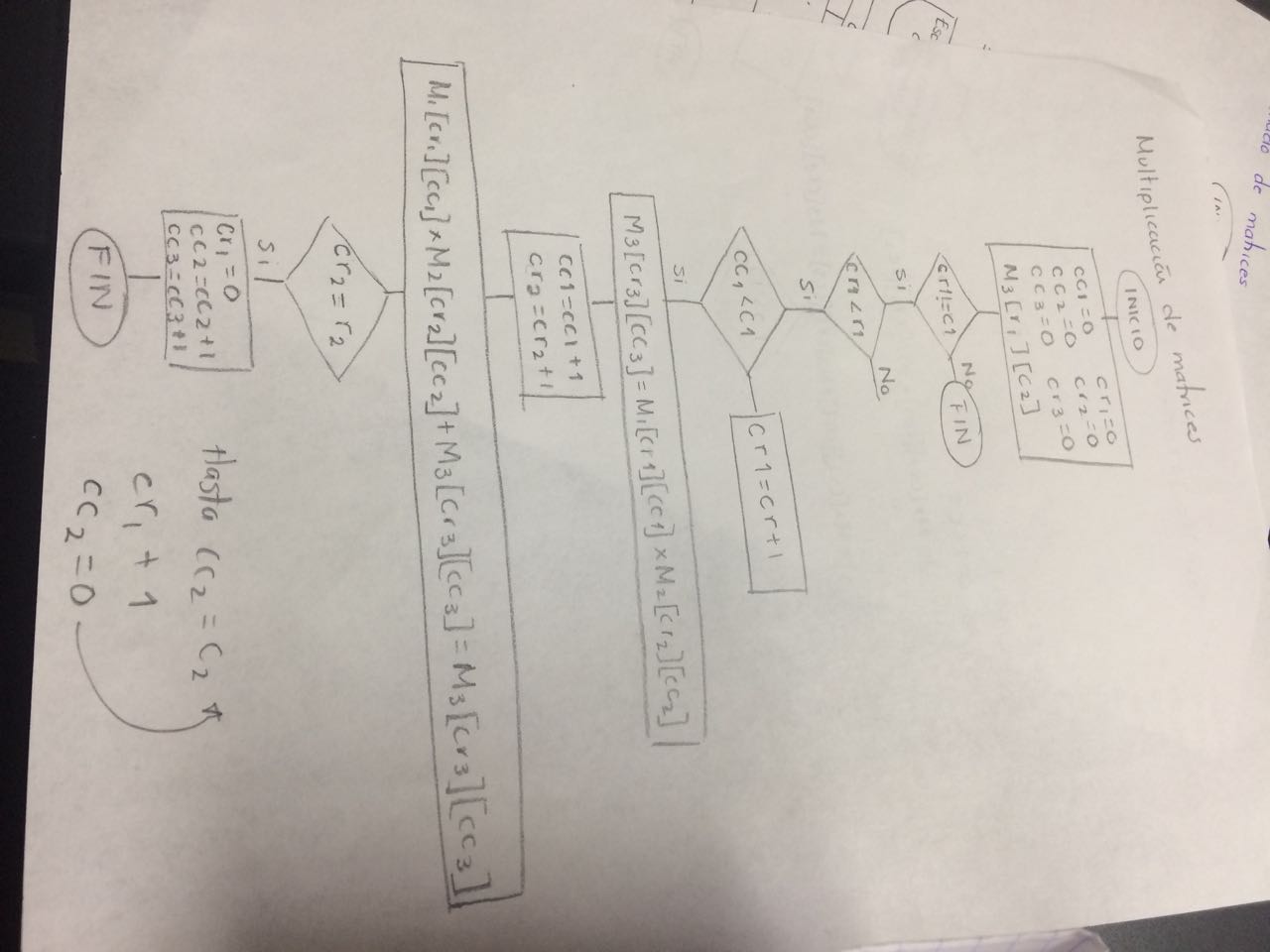
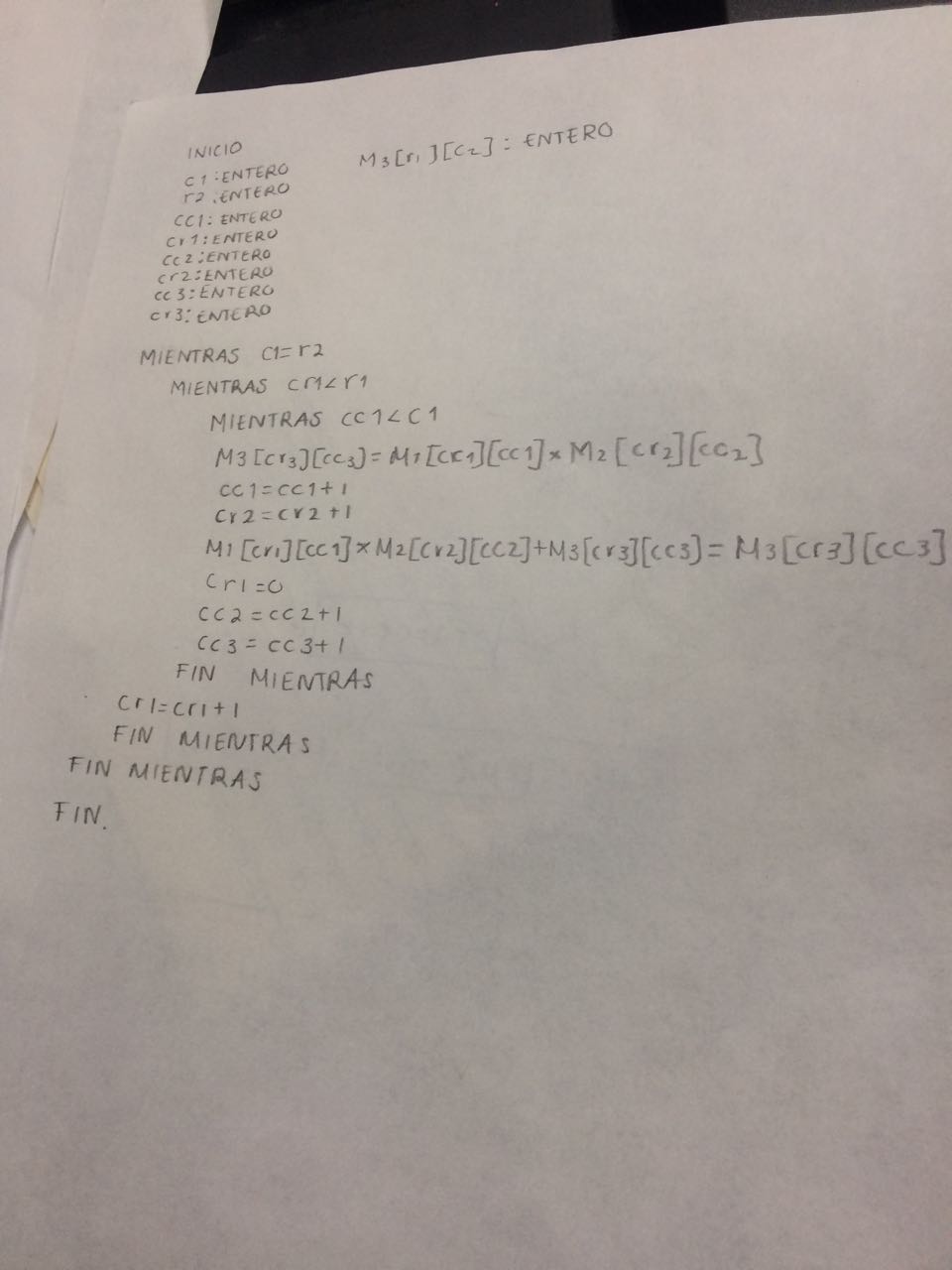
|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

|  |
| --- |
| Cruz carlón Juan Alfredo  Fundamentos de programación  1107  9  Márquez Vázquez lidia Vanesa  2018 - 1  25 de agosto 2017  *Profesor:* |
| *Asignatura:* |
| *Grupo:* |
| *No de Práctica(s):* |
| *Integrante(s):* |
|  |
|  |
| *Semestre:* |
| *Fecha de entrega:* |
| *Obervaciones:* |

salas A y B

****

**multiplicación pseudocodigo**

**LLENADO IMPRECION Y MULTIPLICACION EN C**

#include <stdio.h>

int main()

{//esta llave indica que la funcion principal inicia aqui

int m1[100][100];// matriz 1

int m2[100][100];

int r1;// r1 es el numero ingresado de renglones que tendra la matriz

int r2;//

int c1;//c2 es el numero ingresado de columnas que tendra la matriz

int c2; //

int r; // r es el numero que el usuario va a ingresar en renglones

int c; // c es el numero que el usuario va a ingresar en columnas

printf("escribe el numero de renglones r1 y columnas c1\n");// esto es un mensaje que se envia al usuario, como las instrucciones para que llene el numero de renglones y columnas que quiere que tenga la matriz

scanf("%d",&r1);// lee el numero de reglones que quiere el usuario que tenga la matriz

scanf("%d",&c1);//lee el numero de reglones que quiere el usuario que tenga la matriz

int c\_r;//declaracion de variable para contar renglones

int c\_c;//declaracion de variable para contar columnas

c\_r=0;//asignacion de variable para contador de renglones

c\_c=0;//asignacion de variable para contador de columnas

while (c\_r<r1)//condicion que se ejecutara cuando el contador de renglones sea menor que el numero de renglones que el usuario ingreso

{//esta llave indica que la condicion inicia aqui

c\_c=0;

while(c\_c<c1)//condicion que se ejecutara cuando el contador de columnas sea menor que el numero de columnas que el usuario ingreso

{//esta llave indica que la condicion inicia aqui

printf("Introduce Matriz1 [%d][%d]",c\_r,c\_c); //instruccion que aparecera en la pantalla para que el usuario la lleve a cabo

scanf("%d",&m1[c\_r][c\_c]);//indica que se lee el numero

c\_c=c\_c+1;//indica que el contador de columnas se va a incrementar en una unidad

}//esta llave indica que la condicion termina aqui

c\_r=c\_r+1;//indica que el contador de renglones se va a incrementar en una unidad

}//esta llave indica que la condicion termina aqui

c\_r=0;

c\_c=0;

while(c\_r<r1)

{

while(c\_c<c1)

{

printf("%d", m1[c\_r][c\_c]);

c\_c=c\_c+1;

}

c\_r=c\_r+1;

c\_c=0;

printf("\n");

}

int c\_r2;

int c\_c2;

printf("escribe el numero de renglones r2 y columnas c2\n");// esto es un mensaje que se envia al usuario, como las instrucciones para que llene el numero de renglones y columnas que quiere que tenga la matriz

scanf("%d",&r2);// lee el numero de reglones que quiere el usuario que tenga la matriz

scanf("%d",&c2);//lee el numero de reglones que quiere el usuario que tenga la matriz

c\_r2=0;//asignacion de variable para contador de renglones

c\_c2=0;//asignacion de variable para contador de columnas

while (c\_r2<r2)//condicion que se ejecutara cuando el contador de renglones sea menor que el numero de renglones que el usuario ingreso

{//esta llave indica que la condicion inicia aqui

c\_c2=0;

while(c\_c2<c2)//condicion que se ejecutara cuando el contador de columnas sea menor que el numero de columnas que el usuario ingreso

{//esta llave indica que la condicion inicia aqui

printf("Introduce Matriz2 [%d][%d]",c\_r2,c\_c2); //instruccion que aparecera en la pantalla para que el usuario la lleve a cabo

scanf("%d",&m2[c\_r2][c\_c2]);//indica que se lee el numero

c\_c2=c\_c2+1;//indica que el contador de columnas se va a incrementar en una unidad

}//esta llave indica que la condicion termina aqui

c\_r2=c\_r2+1;//indica que el contador de renglones se va a incrementar en una unidad

}//esta llave indica que la condicion termina aqui

c\_r2=0;

c\_c2=0;

while(c\_r2<r2)

{

while(c\_c2<c2)

{

printf("%d", m2[c\_r2][c\_c2]);

c\_c2=c\_c2+1;

}

c\_r2=c\_r2+1;

c\_c2=0;

printf("\n");

}

printf("el resultado de la multiplicacion de matrices es:");

int m3[100][100];

int c\_r3;

int c\_c3;

c\_c=0;

c\_c2=0;

c\_c3=0;

c\_r=0;

c\_r2=0;

c\_r3=0;

int resultado[100][100];

while (c\_r!=c1)

{

while(c\_r<r1)

{

c\_r=0;

c\_r=c\_r+1;

while(c\_c2<c2)

{

c\_c2=c\_c2+1;

m3[c\_r3][c\_c3]=0;

int k;

while(k<c2)

{

k=0;

k=k+1;

resultado[r1][c2]+=m1[r1][k]\*m2[k][c2];

}

// m1[c\_r][c\_c]\*m2[c\_r2][c\_c2];

// c\_c=c\_c+1;

// c\_r2=c\_r2+1;

// resultado=m1[c\_r][c\_c]\*m2[c\_r2][c\_c2]+m3[c\_r3][c\_c3];

// printf("%d", m3[c\_r3][c\_c3]);

// c\_r=c\_r+1;

// while(c\_r2!=r2)

// {

// c\_r=0;

// c\_c2=c\_c2+1;

// c\_c3=c\_c3+1;

// while(c\_c2!=c2)

// {

// c\_r=c\_r+1;

// c\_c2=0;

//

// }

// }

}

//c\_r=c\_r+1;

printf("%d", m3[c\_r3][c\_c3]);

}

}

c\_r3=0;

c\_c3=0;

int r3;

r3=r1;

int c3;

c3=c2;

//while(c\_r3<r3)

//{

//while(c\_c3<c3)

//{

// printf("%d", m3[c\_r3][c\_c3]);

// c\_c3=c\_c3+1;

//}

//c\_r3=c\_r3+1;

//c\_c3=0;

//printf("\n");

return 0;

//}

}//esta llave indica que la funcion principal termina aqui