



Universidad Nororiental Privada

“Gran Mariscal de Ayacucho”

Facultad De Ingeniera - Ingeniería En Informática y Sistemas

Núcleo Ciudad Guayana

Ejercicios Segundo corte.

Index 0	1	2	3	4	
0	65,340	12,483	138,189	902,960	633,877
1	5,246	424,642	650,380	821,254	866,122
2	89,678	236,781	601,691	329,274	913,534
3	103,902	4,567	733,611	263,010	85,450
4	2,778	658,305	128,788	978,155	620,702
5	45,024	55,058	705,586	89,672	384,605
6	780	47,538	523,784	556,801	617,107
7	32,667	350,890	834,753	638,108	85,188
8	56,083	145,582	775,040	548,322	756,587
9	41,123	543,542	537,738	513,048	418,482

Docente:
Thays Parra

Estudiantes:
Jamal Souki
C.I:31.522.107.

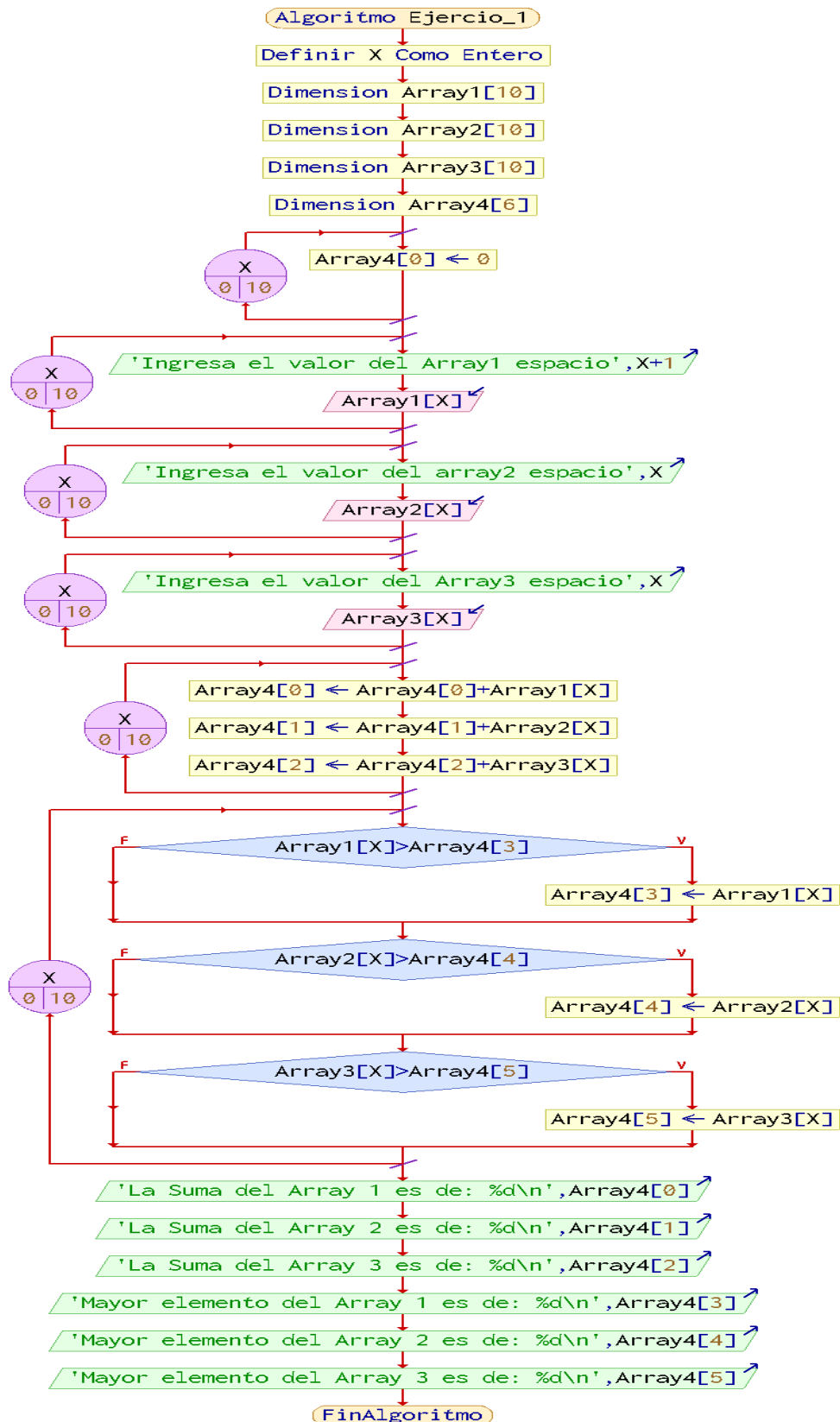
Keysha Montes de Oca.
C.I:30.648.421.

Puerto Ordaz 28 de octubre del 2021

Ejercicio N° 1.

Planteamiento: Dados por pantalla los valores para 3 array de 10 elementos enteros cada uno, genere un cuarto array de 6 posiciones donde las primeras 3 posiciones serán el acumulador de cada uno de los 3 arreglos. Las 3 posiciones deberán contener el mayor elemento de cada uno de los 3 vectores.

Algoritmo General de la solución:



Código de fuente comentado:

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int Array1[9], Array2[9], Array3[9], Array4[5]; //Aqui se encuentran declarados  
    los tres arrays y el cuarto
```

```
    int X=0; //Declaramos la variable
```

```
    for(X; X<10; X++) //Creamos un ciclo para setear a 0
```

```
    {
```

```
        Array4[X]=0; //Aqui seteamos en 0 el cuarto array
```

```
    }
```

```
    X=0;
```

```
    for(X; X<10; X++) //Creamos un ciclo para ir almacenando los valores del array
```

1

```
    {
```

```
        printf("Ingresa el valor del array 1 Espacio: %d\n", X+1); //Aqui muestra un  
        texto señalando el espacio donde se almacenara
```

```
        cin>>Array1[X]; //Guardamos el valor ingresado en el array
```

```
    }
```

```
    X=0;
```

```
    for(X; X<10; X++) //Creamos un ciclo para ir almacenando los valores del array
```

2

```

{
    printf("Ingresa el valor del array 2 Espacio: %d\n", X+1); //Aquí muestra un
    texto señalando el espacio donde se almacenará
    cin>>Array2[X]; //Guardamos el valor ingresado en el array
}

```

```

X=0;

for(X; X<10; X++) //Creamos un ciclo para ir almacenando los valores del array
3

```

```

{
    printf("Ingresa el valor del array 3 Espacio: %d\n", X+1); //Aquí muestra un
    texto señalando el espacio donde se almacenará
    cin>>Array3[X]; //Guardamos el valor ingresado en el array
}

```

```

X=0;

for(X; X<10; X++) //Creamos un ciclo para ir sumando los primeros 3 arreglos y
los guardamos en los primeros 3 espacios del 4 arreglo

```

```

{
    Array4[0] = Array4[0] + Array1[X];
    Array4[1] = Array4[1] + Array2[X];
    Array4[2] = Array4[2] + Array3[X];
}

```

```

X=0;

for(X; X<10; X++) //Creamos un ciclo y usando if vamos viendo si el valor del
espacio anterior es mayor o menor si es mayor

{
    //Guarda en los últimos 3 espacios del 4 array el valor del mayor de
los primeros 3 arrays
}

```

```

    if(Array1[X] > Array4[3])
    {
        Array4[3] = Array1[X];
    }
    if(Array2[X] > Array4[4])
    {
        Array4[4] = Array2[X];
    }
    if(Array3[X] > Array4[5])
    {
        Array4[5] = Array3[X];
    }
}

//Mostramos los resultados del cuarto array
printf("La Suma del Array 1 es de: %d\n", Array4[0]);
printf("La Suma del Array 2 es de: %d\n", Array4[1]);
printf("La Suma del Array 3 es de: %d\n", Array4[2]);

printf("Mayor elemento del Array 1 es de: %d\n", Array4[3]);
printf("Mayor elemento del Array 2 es de: %d\n", Array4[4]);
printf("Mayor elemento del Array 3 es de: %d\n", Array4[5]);

cin.get();
cin.get();
return 1;
}

```

Tabla de Variables usadas:

Variable usada	Tipo de dato de la variable	Función
Array (1,2,3)	Int arreglo	Almacena los datos ingresados por el usuario.
Array 4[5]	Int arreglo	<p>FUNCION 1 :Array 4[0]-[1]-[2] Muestran la sumatoria de los primeros 3 arrays .</p> <p>FUNCION 2 :Array 4[3]-[4]-[5] Almacena dato mayor de los primeros 3 arrays.</p>
X	Int	Contador de los ciclos for en función del programa.

Explicación de la estructura de datos:

Números: entero (int) Almacena enteros sin números decimales.

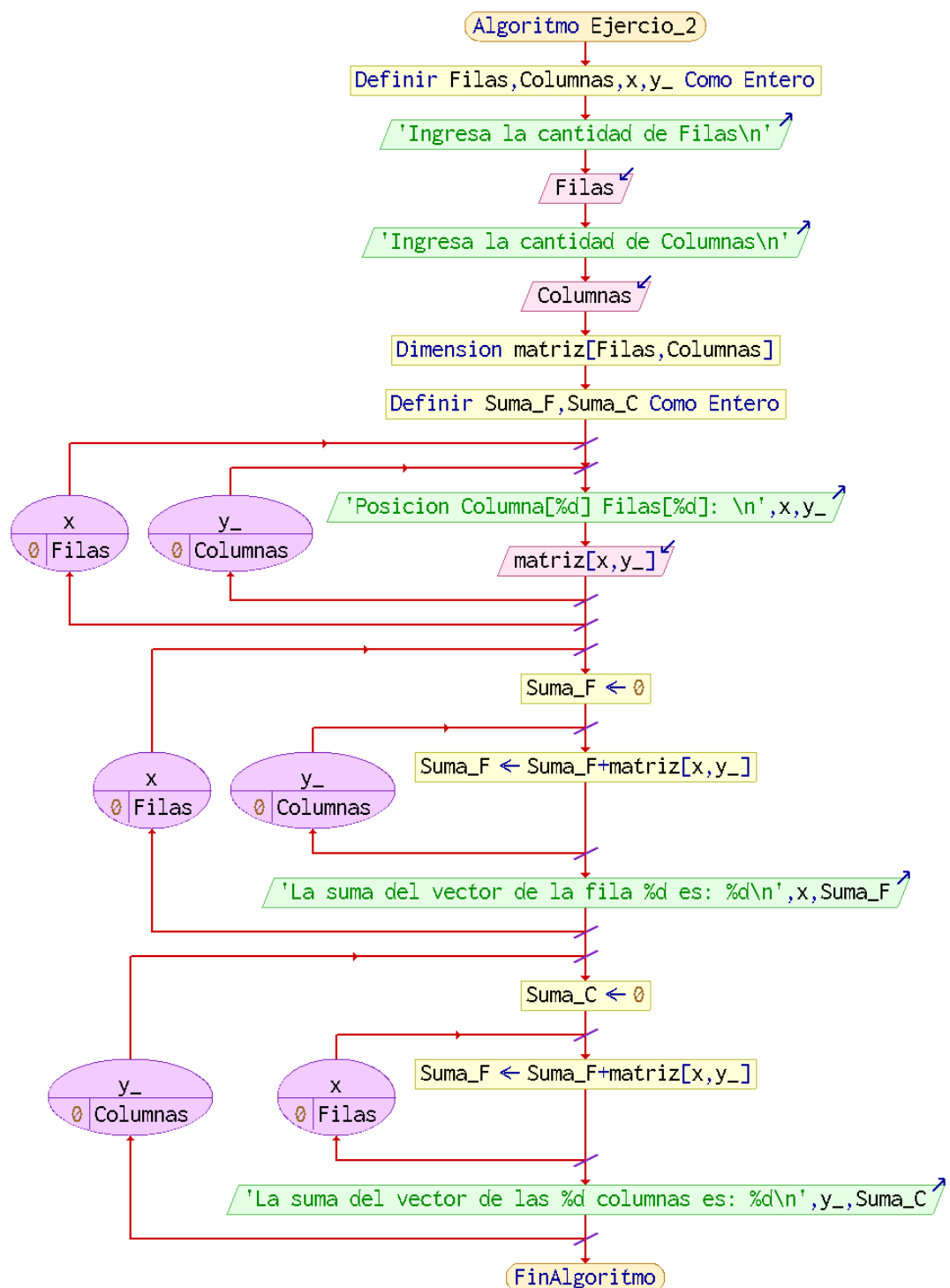
For: Es un ciclo que se ejecuta mientras el código lo apruebe como verdadero este se finaliza cuando se considera falso.

Arrays: Es un tipo de estructura que permite almacenar un conjunto de datos homogéneos.

Ejercicio N° 2.

Planteamiento: Dada una matriz de NxM de valores de enteros, genere 2 vectores: Un vector de tamaño N cuyos elementos estarán conformados por el acumulador de cada una de las filas de la matriz y el otro vector de tamaño M donde sus elementos estarán conformados por el acumulador de cada una de las columnas de la matriz.

Algoritmo General de la solución:



Código de fuente comentado:

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int Filas, Columnas;//Definimos las variables
```

```
    cout<<"Ingresa la cantidad de Filas\n";
```

```
    cin>>Filas;//Guardamos en la variable fila el dato ingresado
```

```
    cout<<"Ingresa la cantidad de Columnas\n";
```

```
    cin>>Columnas;//Guardamos en la variable columnas el dato ingresado
```

```
    int matriz[Filas][Columnas];//Creamos el array de la matriz
```

```
    int Suma_F,Suma_C;//creamos las variables las cuales van a guardar la  
sumatoria
```

```
    int x=0, y=0; //Seteamos 0 las variables x y y
```

```
    for(x=0; x<Filas; x++)//creamos un ciclo para ir almacenando en la matriz los  
valores
```

```
    {
```

```
        for(y=0; y<Columnas ; y++)
```

```
        {
```

```
            printf("Posicion Columna[%d] Filas[%d]: \n", x, y);
```

```
            cin>>matriz[x][y];//Guardamos los valores en la matriz
```



```
}  
}
```

```
for(x=0;x<Filas;x++)//creamos un ciclo para sumar las Columnas  
{  
    Suma_F=0;  
    for(int y=0;y<Columnas;y++)  
    {  
        Suma_F=Suma_F+matriz[x][y];//Guardamos la suma de las columnas  
de la matriz  
    }  
    printf("La suma del vector de la fila %d es: %d\n",x, Suma_F);  
}  
cout<<endl;
```

```
for(int y=0;y<Columnas;y++)//creamos un ciclo para sumar las Filas  
{  
    Suma_C=0;  
    for(int x=0;x<Filas;x++)  
    {  
        Suma_C=Suma_C+matriz[x][y];//Guardamos la suma de las filas de la  
matriz  
    }  
    printf("La suma del vector de las %d columnas es: %d\n", y, Suma_C);//se  
el acumulado de cada una de las columnas de la matriz  
}
```

```
cin.get();
```

```

    cin.get();

    return 1;

}

```

Tabla de Variables usadas:

Variable usada	Tipo de dato de la variable	Función
Filas	Int	Almacena el número de filas.
Columnas	Int	Almacena el número de columnas.
matriz	Int arreglo	Almacena los datos asignados por el usuario.
Suma_f	Int	Almacena los datos de las filas.
Suma_c	Int	Almacena los datos de las columnas .
x	Int	Almacena la cantidad de repeticiones de los ciclos de las filas dadas por el usuario.
y	Int	Almacena la cantidad de repeticiones de los ciclos de las columnas dadas por el usuario.

Explicación de la estructura de datos:

Números: entero (int) Almacena enteros sin números decimales

For: Es un ciclo que se ejecuta mientras el código lo apruebe como verdadero este se finaliza cuando se considera falso.

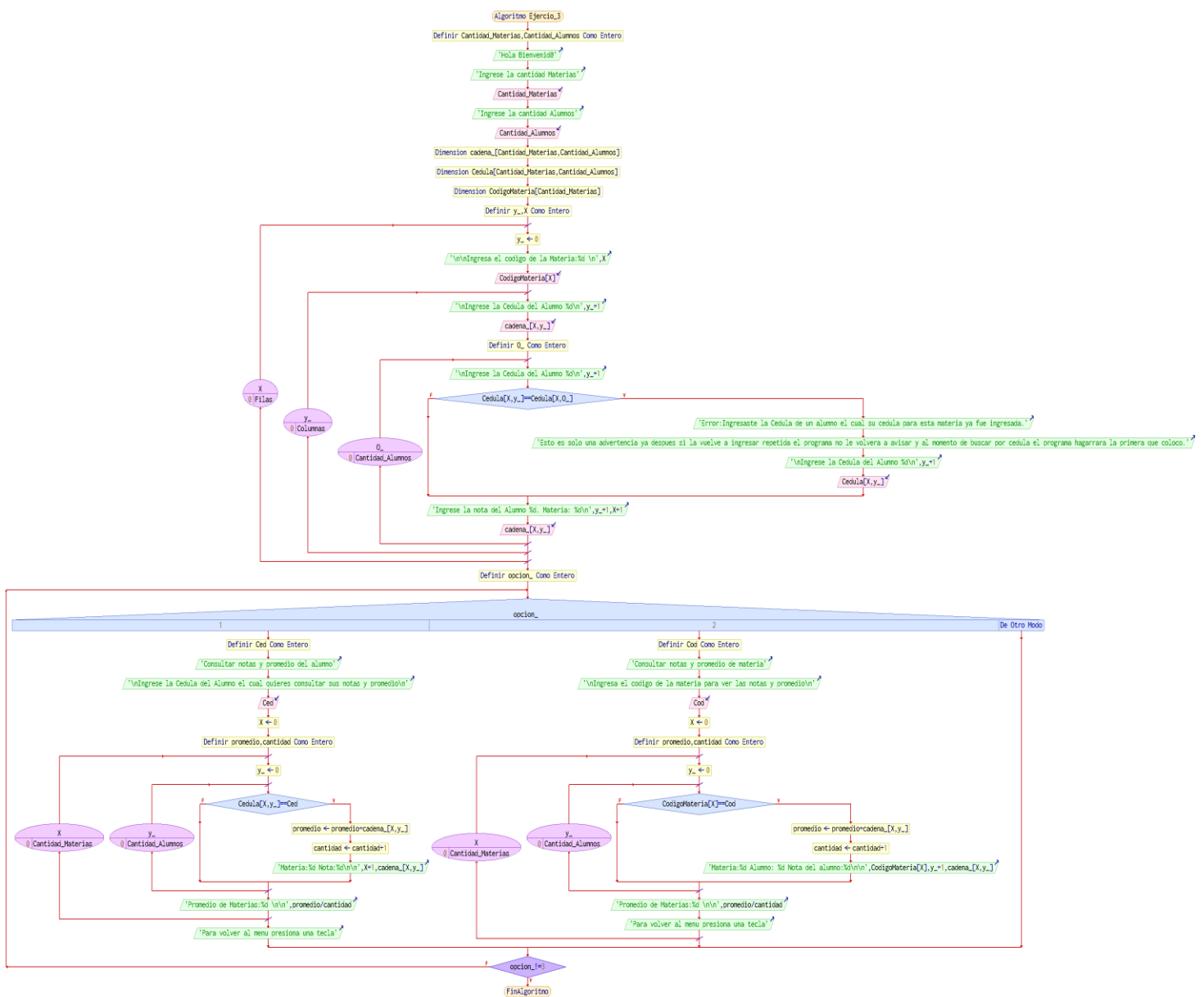
Arrays: Es un tipo de estructura que permite almacenar un conjunto de datos homogéneos..

Ejercicio N° 3.

Planteamiento: Se requiere llevar el registro de notas de N alumnos que cursan M materias. El programa debe permitir:

- A) Consultar las notas y promedio de un alumno introduciendo su Cedula.
- B) Consultar el promedio de notas por materia dando el código de la materia.

Algoritmo General de la solución:



Código de fuente comentado:

```
#include <cstdlib>
```

```
#include <iostream>
```

```
#include <windows.h>
```

```
using namespace std;
```

```
int Cantidad_Materias, Cantidad_Alumnos; //Declaramos las variables  
Cantidad_Materias,Cantidad_Alumnos En estas almacena la cantida de  
materias dada por el usuario y la cantidad de alumnos
```

```
void gotoxy(int x,int y){
```

```
    HANDLE hcon;
```

```
    hcon = GetStdHandle(STD_OUTPUT_HANDLE);
```

```
    COORD dwPos;
```

```
    dwPos.X = x;
```

```
    dwPos.Y= y;
```

```
    SetConsoleCursorPosition(hcon,dwPos);
```

```
}
```

```
int menu(){//Carga el menu Donde muestra las opciones para elegir y retorna  
esa opcion seleccionada
```

```
    int opcion;
```

```
    system("cls");//Borramos el log del programa con system("cls");
```

```
    gotoxy(20,2); cout << "Elija una opcion:";
```

```
    gotoxy(25,4); cout << "1. Consultar notas y promedio de las materias de  
un alumno a través de cedula";
```

```
gotoxy(25,6); cout << "2. Consultar el promedio de notas por materia dando el código de la materia.";
```

```
gotoxy(25,8); cout << "3. Salir";
```

```
gotoxy(28,10); cout << "Opcion: ";
```

```
gotoxy(36,10); cin >> opcion;//Guarda en la variable opcion el valor ingresado
```

```
return opcion;//Retorna la variable opcion
```

```
}
```

```
int main()
```

```
{
```

```
int cadena[Cantidad_Materias][Cantidad_Alumnos];//Esta Array almacena las materias y las notas de los alumnos
```

```
int Cedula[Cantidad_Materias][Cantidad_Alumnos];//Esta Array almacena el numero de materia y cedulas por materias diferentes
```

```
int CodigoMateria[Cantidad_Materias];//Este Array almacena el CodigoMateria
```

```
gotoxy(31,2); cout <<"Hola Bienvenid@\n\n"<<endl;//Da el mensaje de bienvenida
```

```
//Le solicita los datos al usuario de la cantidad de materias y la cantidad de alumnos
```

```
cout<<"Ingrese la cantidad Materias"<<endl;
```

```
cin>>Cantidad_Materias;//Guarda la cantidad de materias insertada
```

```
cout<<"Ingrese la cantidad Alumnos"<<endl;
```

```
cin>>Cantidad_Alumnos;//Guarda la cantidad de alumnos insertada
```

```
int y=0,X=0;//Creamos y ponemos los contados y X en 0
```

for(X;X<Cantidad_Materias;X++)//creamos un ciclo para las Filas las cuales son la Cantidad_Materias

{

y =0;//Seteamos la columna actual en 0

printf("\n\nIngresa el codigo de la Materia:%d \n",X+1);

cin>>CodigoMateria[X];//Guarda el codigo de la materia

for(y;y<Cantidad_Alumnos; y++)//Creamos un ciclo para las columnas Donde almacena las notas y cedula de los alumnos

{

printf("\nIngresa la Cedula del Alumno %d\n",y+1);

cin>>Cedula[X][y];//Guardamos la cedula del alumno

cout<<"\n";

for(int O=0;O<Cantidad_Alumnos; O++)//Este ciclo es para comparar si la cedula ya fue ingresada en la materia anteriormente

{

if(Cedula[X][y] == Cedula[X][O] && y!=O)//Detecta si el alumno ya fue ingresado anteriormente para evitar tener varios con la misma cedula. Sin esto al momento de realizar la busqueda por cedula el programa en caso del ciclo que realiza la busqueda desde 0 en adelante este detectaria solo el mas cerca a 0

{

printf("Error:Ingresaste la Cedula de un alumno el cual su cedula para esta materia ya fue ingresada.");

printf("Esto es solo una advertencia ya despues si la vuelve a ingresar repetida el programa no le volvera a avisar y al momento de buscar por cedula el programa hagarra la primera que coloco.");

printf("\nIngresa la Cedula del Alumno %d\n",y+1);

cin>>Cedula[X][y];//Sobre escribe la cedula del Alumno

cout<<"\n";

```

        break;
    }
}

printf("Ingrese la nota del Alumno %d. Materia: %d\n",y+1,X+1);

cin>>cadena[X][y];//Guardamos la nota del alumno
}
}

int opcion;//Declaramos la variable opcion en donde se almacena la
opcion seleccionada del usuario

opcion = menu();//Ejecutamos la funcion de menu y de hay este retorna en la
variable opcion

do{

    switch(opcion)//creamos un switch el cual devuelve
dependiendo del valor de la variable opcion

    {

        case 1:

        {

            int Ced;//Creamos una variable para
almacenar el numero de cedula dado por el usuario

            system("cls");//Limpiamos el log del programa

            gotoxy(20,2); cout << "Consultar notas y promedio del alumno";

            printf("\nIngrese la Cedula del Alumno el cual quieres consultar
sus notas y promedio\n");

            cin>>Ced;//Guardamos en la variable Ced el numero de la cedula


            X=0;//Seteamos X en 0

            int promedio=0,cantidad=0;//Declaramos 2 variables promedio y
cantidad y las seteamos en 0

            for(X;X<Cantidad_Materias;X++)//Creamos un ciclo para ir por las
filas

```

```

        {
            y = 0; // Seteamos y en 0
            for(y; y < Cantidad_Alumnos; y++) // creamos un ciclo para ir por
las columnas
            {
                if(Cedula[X][y] == Ced) // Vamos comprando con las cedula
ingresadas por otros alumnos y si es igual a la cedula solicitada
                {
                    promedio = promedio + cadena[X][y]; // Vamos sumando la
nota del alumno y la guardamos en promedio
                    cantidad = cantidad + 1; // sumamos 1 por cada materia que
este el alumno (Cada vez que el ciclo se repite)
                    printf("Materia: %d          Nota: %d\n\n",          X+1,
cadena[X][y]); // Mostramos El numero de la materia y La nota del alumno
                }
            }
        }

        printf("Promedio          de          Materias: %d
\n\n", promedio/cantidad); // Dividimos la suma de las notas y la dividimos por la
cantidad de notas para sacar el promedio

```

```

        cout << "Para volver al menu presiona una tecla";

        cin.get();

        cin.get();

        opcion = menu();

                break;

    }

        case 2:

        {

```



```

        int Cod;//Creamos una variable para
almacenar el codigo de la materia dado por el usuario

        system("cls");//Limpiamos el log del programa

        gotoxy(20,2); cout << "Consultar notas y promedio de materia";

        printf("\nIngresa el codigo de la materia para ver las notas y
promedio\n");

        cin>>Cod;//Guardamos en la variable Cod el codigo de la materia

        X=0;//seteamos la variable x a 0

        int promedio=0,cantidad=0;//Declaramos las variables promedio y
cantidad y seteamos en 0

        for(X;X<Cantidad_Materias;X++)//Creamos un ciclo para ir por filas
        {
            y =0;//Seteamos y en 0

            for(y;y<Cantidad_Alumnos; y++)//creamos un ciclo para ir por
las columnas
            {
                if(CodigoMateria[X] == Cod)//Vamos comprando con los
codigos de las materias con la ingresada por el usuario
                {
                    promedio = promedio+cadena[X][y];//SUMamos las notas
                    cantidad = cantidad+1;//Y vamos guardando la cantidad de
notas

                    printf("Materia:%d Alumno: %d Nota del alumno:%d\n\n",
CodigoMateria[X],y+1, cadena[X][y]);
                }
            }
        }

        printf("Promedio de la Materia:%d
\n\n",promedio/cantidad);//Dividimos la suma de las notas y la dividimos por la
cantidad de notas para sacar el promedio

```

```

        cout<<"Para volver al menu presiona una tecla";

        cin.get();

        cin.get();

        opcion = menu();

                                break;

    }

        }

    }while(opcion != 3);

    cin.get();

    cin.get();

    return 1;

}

```

Tabla de Variables usadas:

Variable usada	Tipo de dato de la variable	<u>Función</u>
Cantidad_Materias	Int arreglo	Almacena la cantidad de materias registradas por el usuario.
Cantidad_Alumnos	Int arreglo	Almacena la cantidad de alumnos registradas por el usuario.
CodigoMateria	Int arreglo	Almacena el código de las materias registradas por el usuario.
X	<u>Int</u>	Almacena la cantidad de repeticiones de los ciclos de las filas dadas por el usuario.
Y	<u>Int</u>	Almacena la cantidad de repeticiones de los ciclos de las columnas dadas por el usuario.

O	<u>Int</u>	Almacena la cantidad de repeticiones de los ciclos de las filas dadas por el usuario.
Opcion	<u>Int</u>	Almacena las selecciones del usuario.
Ced	<u>Int</u>	Almacena la cedula dada por el usuario.
promedio	<u>Int</u>	Almacena el cálculo del promedio de las notas.
Cod	<u>Int</u>	Almacena el código de la materia
cantidad	<u>Int</u>	Guarda el número de notas.

Explicación de la estructura de datos:

Números: entero (int) Almacena enteros sin números decimales.

For: Es un ciclo que se ejecuta mientras el código lo apruebe como verdadero este se finaliza cuando se considera falso.

While: Es un bucle que se repite la cantidad de veces mientras se considere verdadero.