

# INTRODUCCIÓ A LA PROGRAMACIÓ ESTRUCTURADA

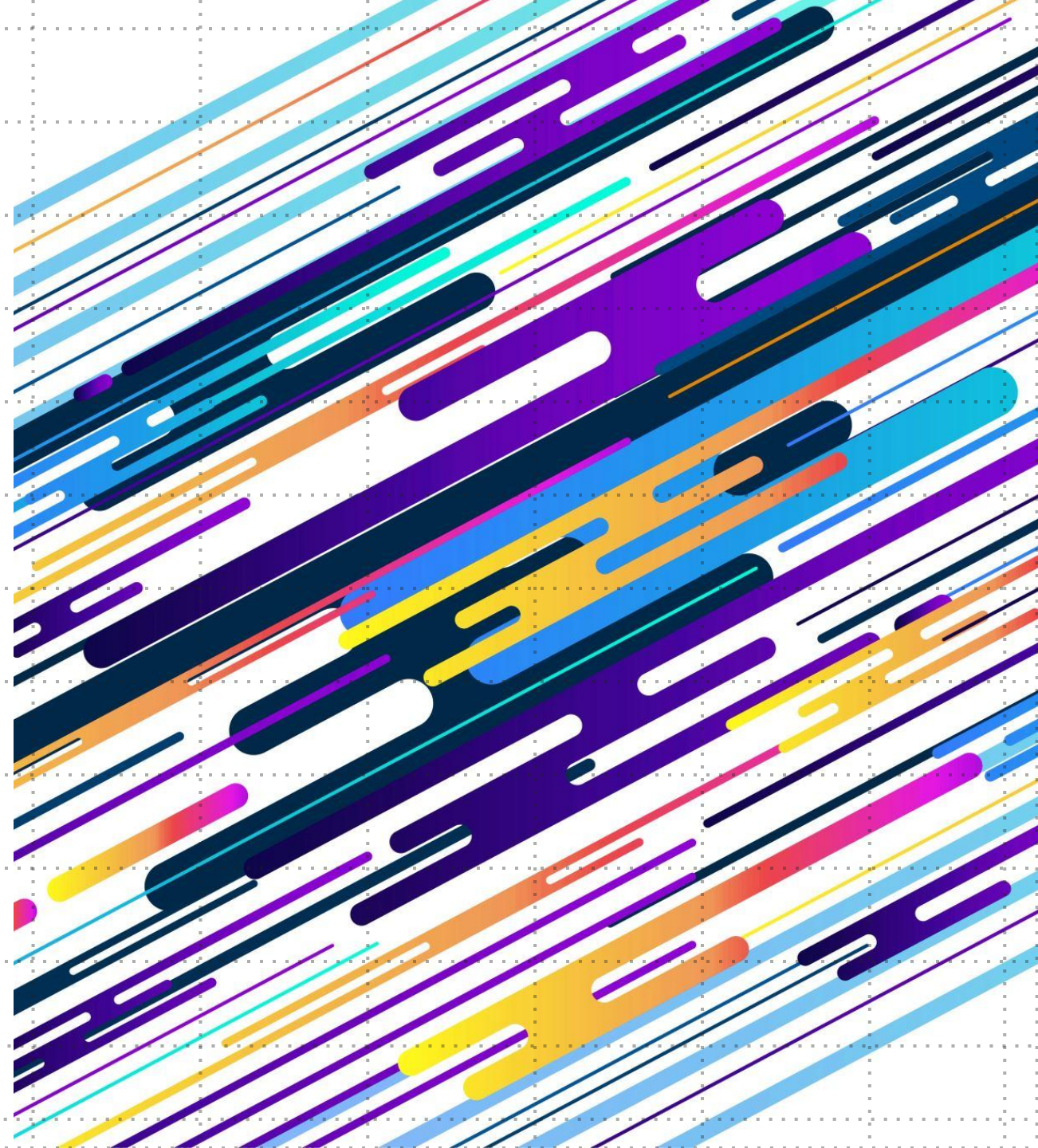
ARRAIGS, ESTRUCTURES I CADENES DE  
TEXT

CFGs DAW

MP03 - UF1

Marc Callejón Beltrán

ETP Xavier



## ARRANJAMENTS

Codifica un programa que introduint la part numèrica del DNI ens digui la lletra.

Calcular la lletra que fa de dígit de control és molt fàcil, simplement cal fer un mòdul del número 23 i amb el residu calcular el dígit de control mitjançant la taula següent

RESTO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
LETRA	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

## ARRANJAMENTS BI-DIMENSIONALS

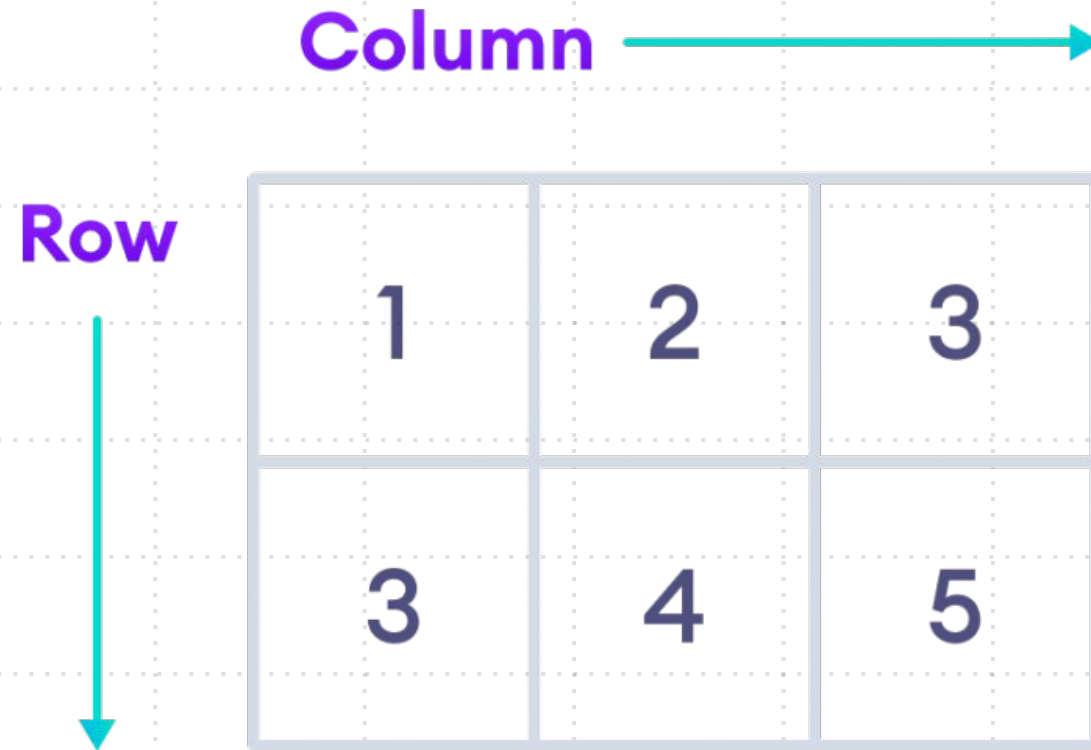
Podem declarar taules de dues dimensions.

Per exemple, si un professor vol desar dades de dos grups d'alumnes, cadascun dels quals té 20 persones, tindria dues opcions:

Es poden fer servir `int dadesAlumnes[40]` i aleshores hem de recordar que les 20 primeres dades corresponen realment a un grup d'alumnes i les 20 següents a un altre grup.

O bé podem emprar **`int dadesAlumnes[2,20]`** i aleshores sabem que les dades de la forma **`dadesAlumnes[0,i]`** són les del primer grup, i les **`dadesAlumnes[1,i]`** són les del segon.

# ARRANJAMENTS BI-DIMENSIONALS



## ARRANJAMENTS BI-DIMENSIONALS

```
int[,] notas1 = new int[2,3]; // 2 filas y 3 columnas
```

```
notas1[0,0] = 1;
```

```
notas1[0,1] = 2;
```

```
notas1[0,2] = 3;
```

```
notas1[1,0] = 4;
```

```
notas1[1,1] = 5;
```

```
notas1[1,1] = 6;
```

```
int[ , ] notas = {{ 1, 2, 3}, { 4, 5, 6 }};
```

# ARRANJAMENTS BI-DIMENSIONALS

//2D array

```
int[ , ] x = { { 1, 2 ,3}, { 3, 4, 5 } };
```

// access al primer element de la primera columna

```
x[0, 0]; // returns 1
```

// access tercer element de la segona

```
x[1, 2]; // returns 5
```

// access tercer element de la primera

```
x[0, 2]; // returns 3
```



## ARRANJAMENTS BI-DIMENSIONALS

```
int[,] notas1 = new int[2,3]; // 2 bloques de 2 datos
```

```
notas1[0,0] = 1;
```

```
notas1[0,1] = 2;
```

```
notas1[0,2] = 3;
```

```
notas1[1,0] = 4;
```

```
notas1[1,1] = 5;
```

```
notas1[1,1] = 6;
```

```
//int[ , ] notas = {
```

```
    { 1, 2, 3},
```

```
    { 4, 5, 6 }
```

```
};
```

```
Console.WriteLine("La nota1 del segundo alumno del grupo 1 es {0}",  
notas1[0,1]);
```

```
Console.WriteLine("La nota2 del tercer alumno del grupo 1 es {0}",  
notas2[0,2]);
```

## ARRANJAMENTS BI-DIMENSIONALS

```
int[,] notas = { { 1, 2, 3 }, { 4, 5, 6 } };
```

```
for (int i = 0; i < 2; i++)
```

```
{
```

```
    Console.WriteLine("Fila: " + i );
```

```
    for (int j = 0; j < 3; j++)
```

```
    {
```

```
        Console.WriteLine("columna: " + j + "nota: " + notas[i, j]);
```

```
    }
```

```
}
```

```
Console.WriteLine();
```



## ARRANJAMENTS BI-DIMENSIONALS

Codifica un programa que ompli un array de 3x3 i posteriorment printi els valors per consola

## ARRANJAMENTS BI-DIMENSIONALS

Suma la primera filera de la matriu

## ARRANJAMENTS BI-DIMENSIONALS

Suma les fileres de l'exercici anterior

## ARRANJAMENTS BI-DIMENSIONALS

Suma les columnes de l'exercici anterior

## ARRANJAMENTS BI-DIMENSIONALS

Codifica un programa que sol·liciti a l'usuari les notes de 10 alumnes dividits en 2 grups de 5 alumnes cadascun. Posteriorment caldrà treure la nota mitjana de cada grup

## ARRANJAMENTS BI-DIMENSIONALS

Codifica un programa amb una matriu bi-dimensional on preguntem a l'usuari el nombre de fileres, de columnes i lletra a guardar, per pintar-la després.

## ARRANJAMENTS BI-DIMENSIONALS

Codifica un arreglo bidimensional que almacena la cantidad de computadores vendidos por tres vendedores en cuatro zonas diferentes.

Se pide mostrar:

El total por zona

El total por vendedor