



# ARRAYS

# INTRODUCCION

Oscar Paniagua @2023

# VARIABLES

✓ Primitive Types in C

✓ Pointers

→ Arrays

# Tipos de datos (repaso)

- Enteros
  - `char`, `int`
- Decimales
  - `float`, `double`
- Modificadores
  - `short` [`int`]
  - `long` [`int`, `double`]
  - `signed` [`char`, `int`]
  - `unsigned` [`char`, `int`]

32 ó 64 bit

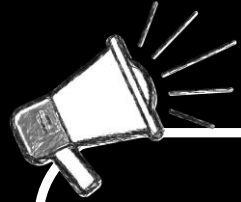
```
$ arch
$ echo $MACHTYPE
$ lscpu
```

| C Data Type                     | 32-bit | 64-bit | printf               |
|---------------------------------|--------|--------|----------------------|
| <code>char</code>               | 1      | 1      | <code>%c</code>      |
| <code>short int</code>          | 2      | 2      | <code>%hd</code>     |
| <code>unsigned short int</code> | 2      | 2      | <code>%hu</code>     |
| <code>int</code>                | 4      | 4      | <code>%d / %i</code> |
| <code>unsigned int</code>       | 4      | 4      | <code>%u</code>      |
| <code>long int</code>           | 4      | 8      | <code>%ld</code>     |
| <code>long long int</code>      | 8      | 8      | <code>%lld</code>    |
| <code>float</code>              | 4      | 4      | <code>%f</code>      |
| <code>double</code>             | 8      | 8      | <code>%lf</code>     |
| <code>long double</code>        | 12     | 16     | <code>%Lf</code>     |
| <code>pointer</code>            | 4      | 8      | <code>%p</code>      |

# A programar...

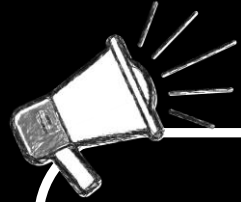
>\_

- a) Ingrese 3 valores enteros y calcule el promedio. Los datos deberán ser ingresados por teclado.
- b) Modifique el programa anterior para que se muestren los valores ingresados mayor al promedio
- c) Modifique el programa anterior para que se puedan ingresar hasta 99 valores



## array@definición

- Es una “estructura de datos” que “contiene un conjunto de datos” (valores) del “mismo tipo”, en general se lo suele llamar también “colección”



## array@definición

- Es una “estructura de datos” que “contiene un conjunto de datos” (valores) del “**mismo tipo**”, en general se lo suele llamar también “colección”

# “ESTRUCTURA DE DATOS”

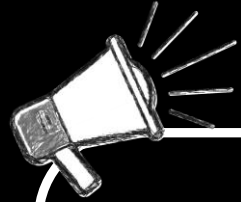
Es una forma particular de **organizar** y **almacenar** información para que pueda ser utilizada de manera eficiente.

Existen varios tipos de estructuras de datos, cada uno de ellos responden a un uso específico

**Ejemplo:** “array”



porción de memoria **organizada** de manera secuencial, dividida en bloques de memoria pequeños, capaz de **almacenar** un valor de un cierto tipo



## array@definición

- Es una “estructura de datos” que “contiene un conjunto de datos” (valores) del “**mismo tipo**”, en general se lo suele llamar también “colección”



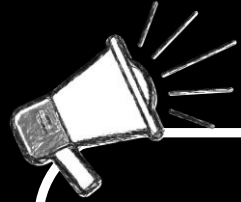
# UN ARRAY CONTIENE UN CONJUNTO DE DATOS

Este array contiene 5 valores int(enteros)

|     |   |   |    |   |   |
|-----|---|---|----|---|---|
| arr | 3 | 7 | 11 | 1 | 9 |
|-----|---|---|----|---|---|

Este array contiene 11 valores int(enteros)

|     |   |   |    |   |   |   |    |    |    |    |   |
|-----|---|---|----|---|---|---|----|----|----|----|---|
| arr | 3 | 7 | 11 | 1 | 9 | 0 | -1 | 11 | 15 | 21 | 9 |
|-----|---|---|----|---|---|---|----|----|----|----|---|



## array@definición

- Es una “estructura de datos” que “contiene un conjunto de datos” (valores) del “mismo tipo”, en general se lo suele llamar también “colección”

# TIPO DE DATOS DE LOS ELEMENTOS DEL ARRAY

a

|   |   |    |   |   |
|---|---|----|---|---|
| 3 | 7 | 11 | 1 | 9 |
|---|---|----|---|---|



b

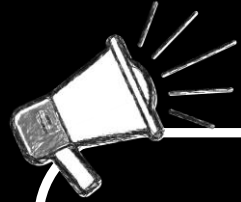
|     |     |     |     |
|-----|-----|-----|-----|
| 'a' | 'b' | 'c' | 'd' |
|-----|-----|-----|-----|



c

|   |     |      |     |    |
|---|-----|------|-----|----|
| 3 | 'a' | 3.14 | 'x' | -1 |
|---|-----|------|-----|----|



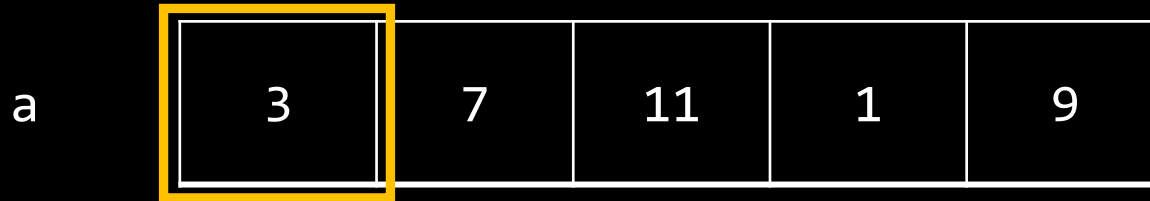


## array@definición

- Es una “estructura de datos” que “contiene un conjunto de datos” (valores) del “mismo tipo”, en general se lo suele llamar también “colección”

# array de una dimensión 1D

El array mas simple que podemos imaginar es de una dimensión....1D



- un array es un bloque grande de memoria
- podemos verlo como una “fila”, dividida en pequeños bloques
- donde cada bloque es capaz de almacenar un valor
- y todos los valores tienen que ser del mismo tipo
- nos podemos imaginar que un bloque del array es una variable
- y el array es una colección de variables