

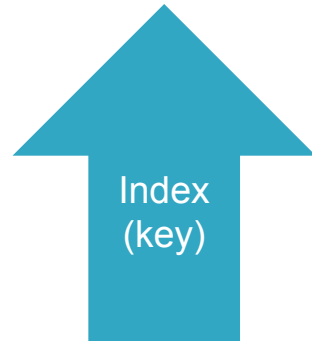


Arreglos

¿Qué es un array?

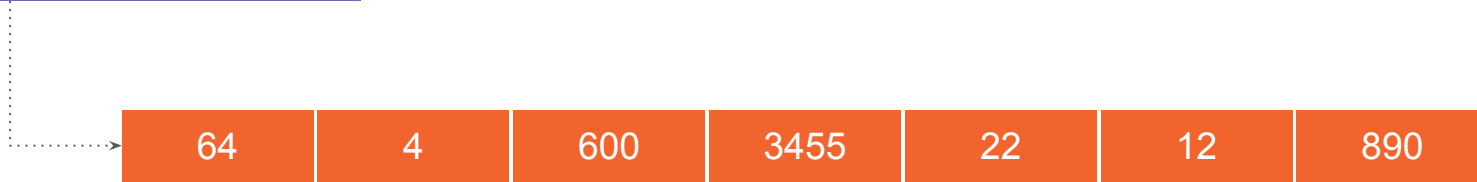


¿Qué es un array?



¿Qué es un array?

Array of integers



¿Qué es un array?

64	4	600	3455	22	12	890
0	1	2	3	4	5	6

Indexes
(0-based)

¿Qué es un array?

64	4	600	3455	22	12	890
0	1	2	3	4	5	6



Ejemplos de utilización de arrays

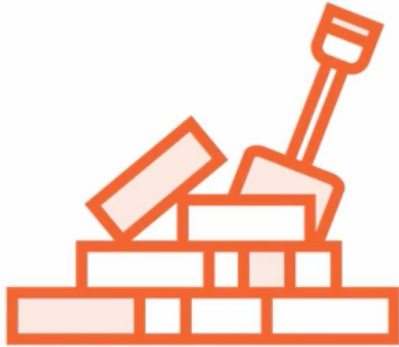


**Lista de
precios**

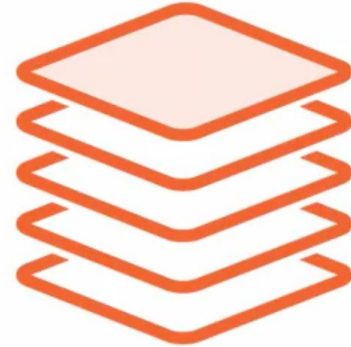


Imágenes

Crear estructuras más complejas



Utilizar arrays como bloques para construir estructuras



Pilas de estructuras de datos

Propiedades de los arrays



Propiedades de los arrays



Propiedades de los arrays

int	int	int	int	int	int	int
-----	-----	-----	-----	-----	-----	-----

float	float	float	float	float	float	float
-------	-------	-------	-------	-------	-------	-------

Propiedades de los arrays

< Homogeneidad > = mismo tipo



int	int	int	int	int	int	int
-----	-----	-----	-----	-----	-----	-----



float	float	float	float	float	float	float
-------	-------	-------	-------	-------	-------	-------



int	float	int	float	int	float	int
-----	-------	-----	-------	-----	-------	-----

Propiedades de los arrays

Acceso rápido a los elementos mediante el índice



a : nombre del array
 i : índice del elemento

Propiedades de los arrays

Acceso rápido a los elementos mediante el índice

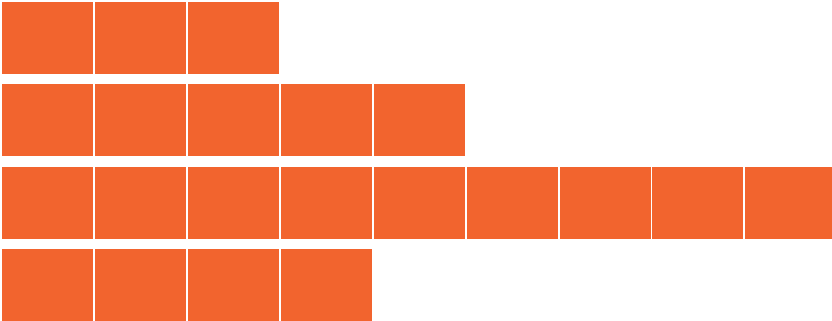


$O(1)$
Constant



Propiedades de los arrays

Acceso rápido a los elementos mediante el índice



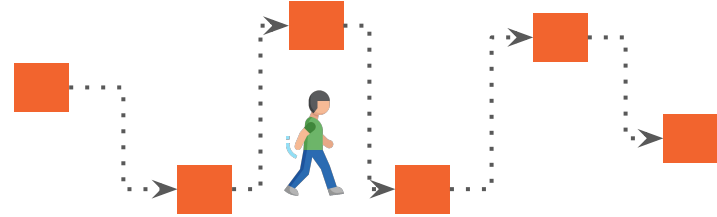
$O(1)$
Constant



Acceso a elementos: Arrays vs Listas enlazadas

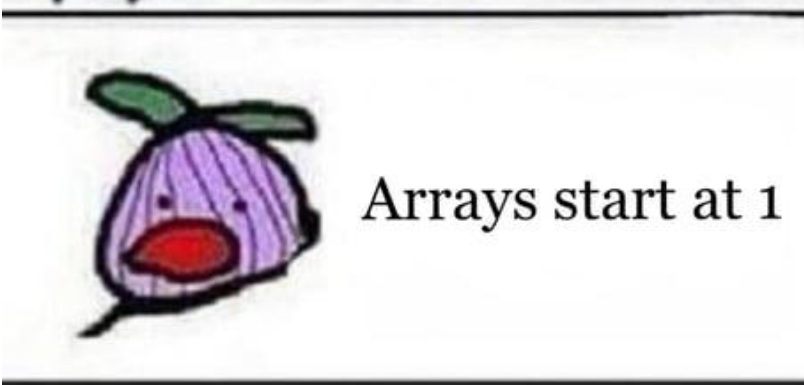
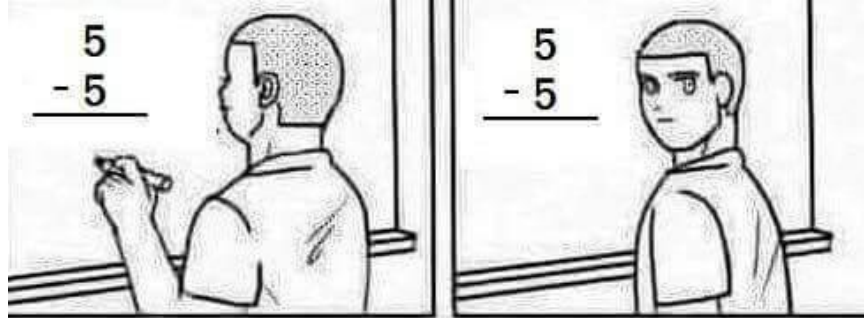


$O(1)$
Constant



$O(N)$
Linear





Resumen Vectores



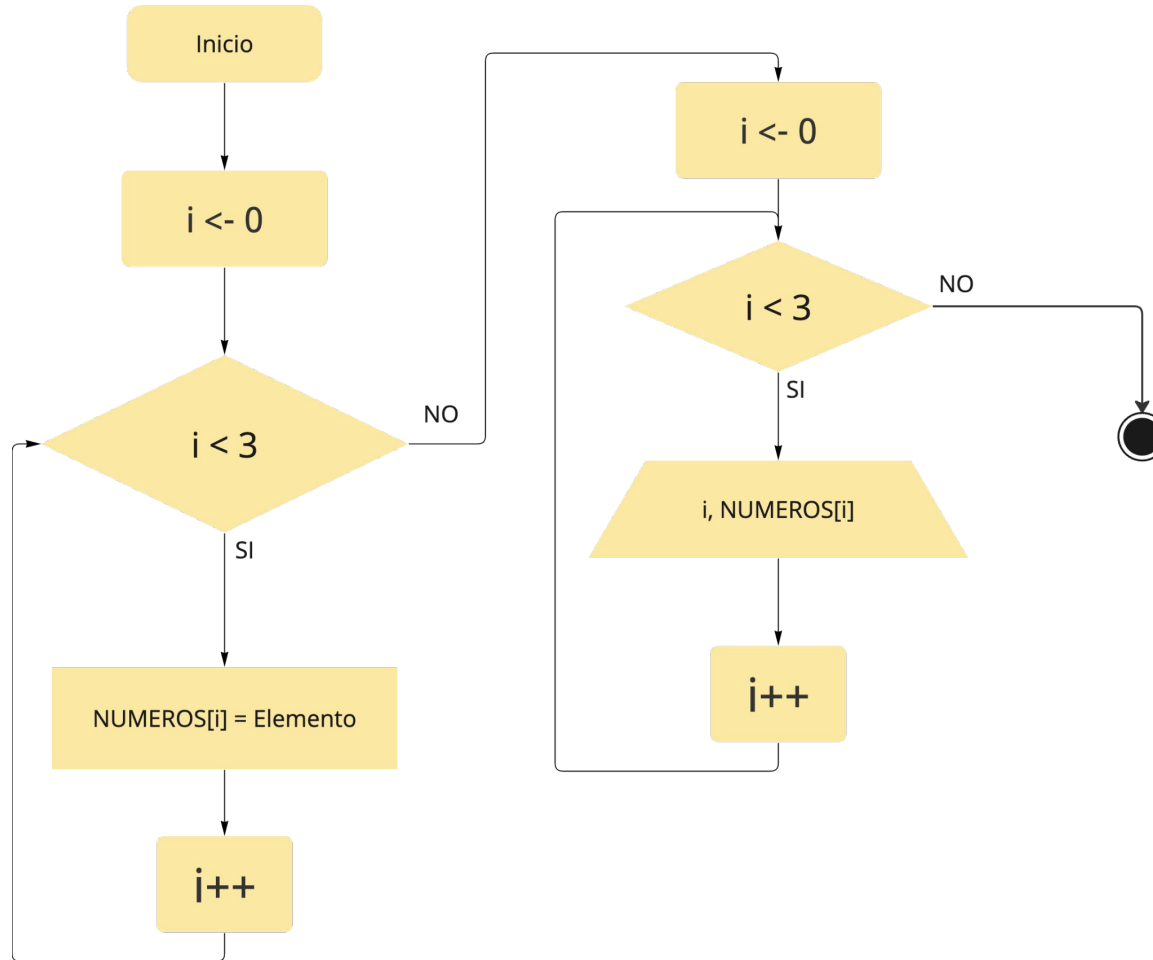
- Los arreglos (arrays) son una agrupación de variables del mismo tipo bajo un único nombre identificador.
- Pueden tener varias **dimensiones**, el más común llamado **vector**
- Muchas estructuras de datos como lo son listas, pilas, colas, árboles, etc pueden implementarse con vectores.

Imprimir el contenido de un array



```
[ 44 2 379 ]
```

Imprimir el contenido de un array



Cadenas de caracteres (“strings”)

char CAD1[8]

M	E	N	S	A	J	E	“\0”
0	1	2	3	4	5	6	7

Cadenas de caracteres (“strings”)

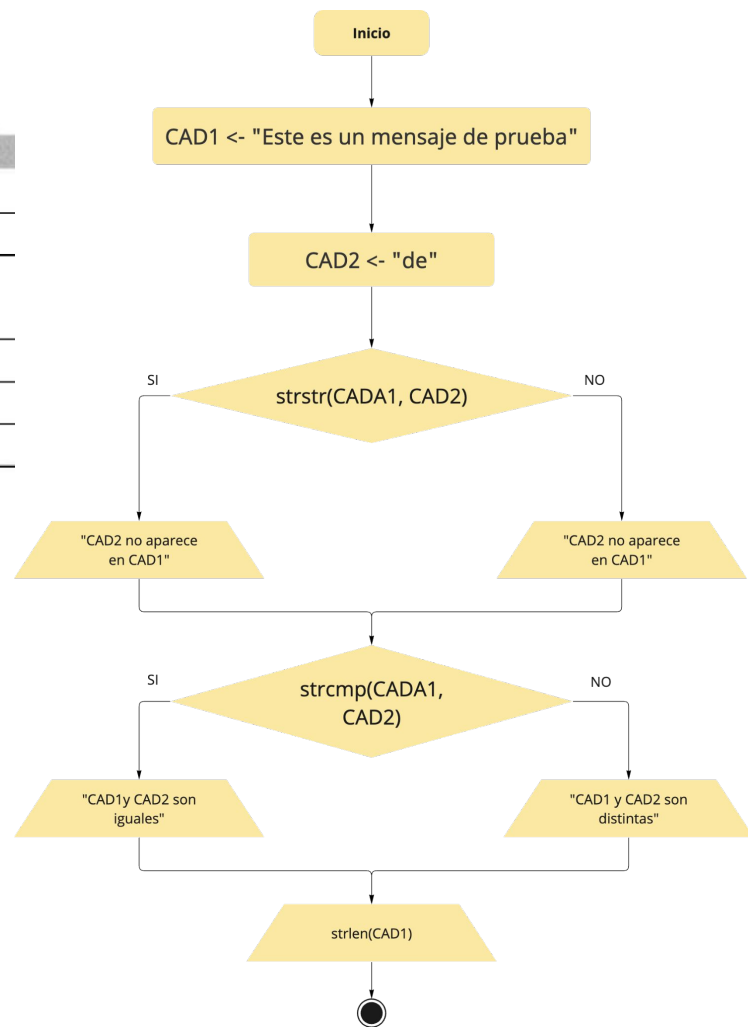
char CAD1[8]



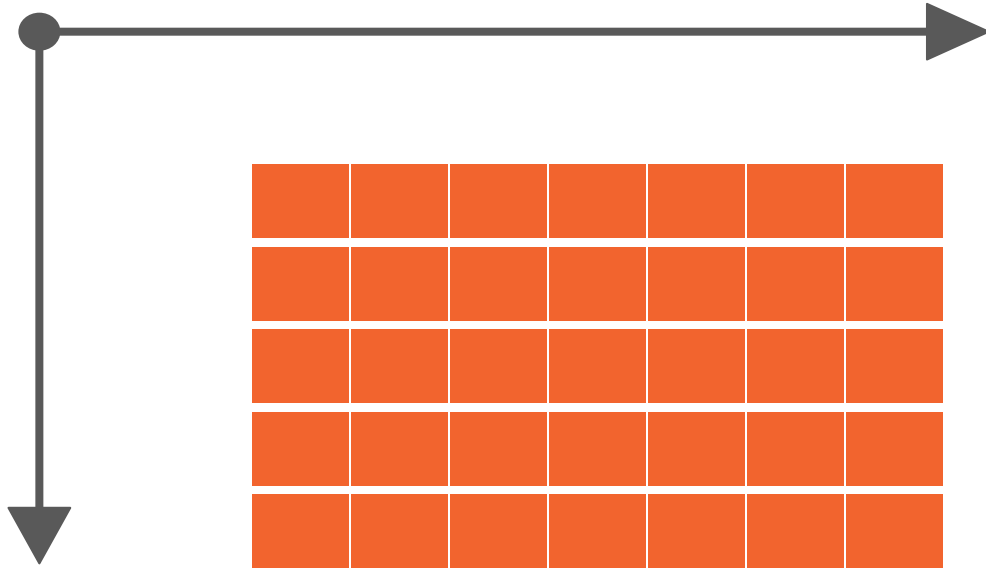
FUNCIÓN	DESCRIPCIÓN
strcpy (c1, c2)	Copia c2 en c1
strlen (c1)	Devuelve la longitud de c1
strcmp (c1, c2)	Compara c1 con c2. Si son iguales, devuelve 0, si c1 > c2, devuelve mayor que 0, y si c1 < c2, devuelve menor que 0.
strcat (c1, c2)	Concatena c2 al final de c1
strstr (c1, c2)	Devuelve un puntero a la primera aparición de c2 en c1
strchr (c1, car)	Devuelve un puntero a la primera aparición de car en c1

Tabla 1. Funciones para el manejo de cadenas de caracteres en C.

- Las cadenas de caracteres se implementan con un vector de tipo **char** donde el último elemento de la cadena debe ser un carácter “\0” (**vacío**) es por esta razón que no se puede asignar una constante de tipo **string** a un cadena de caracteres con el operador de asignación.

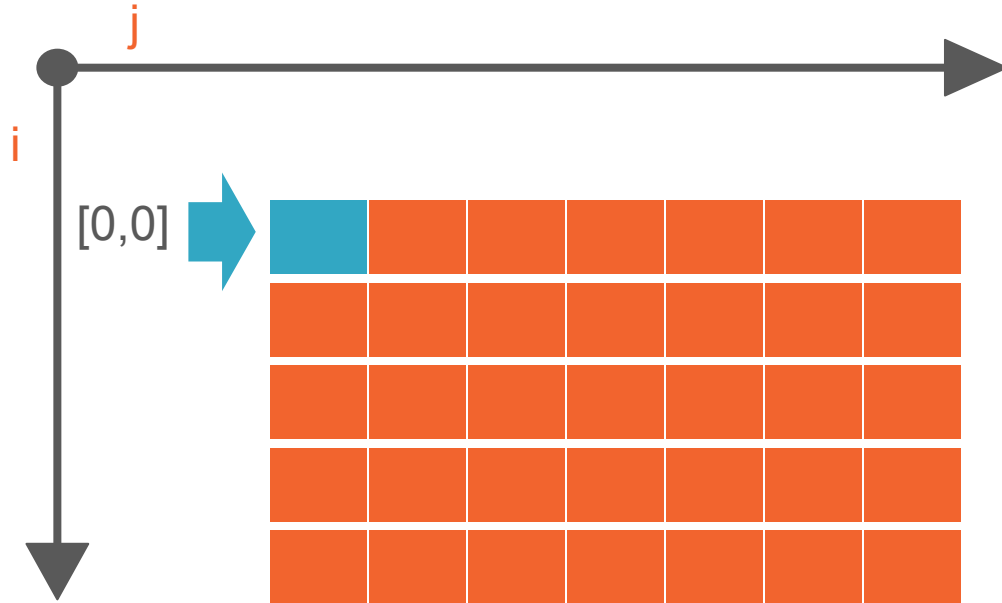


Matrices

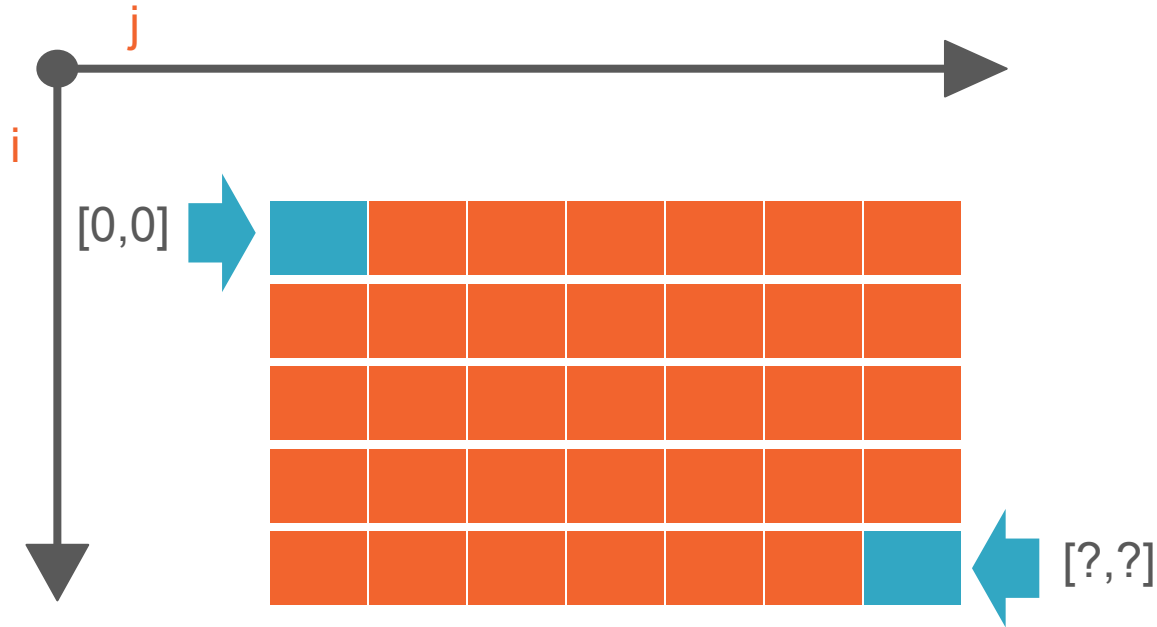


Arrays de 2
dimensiones

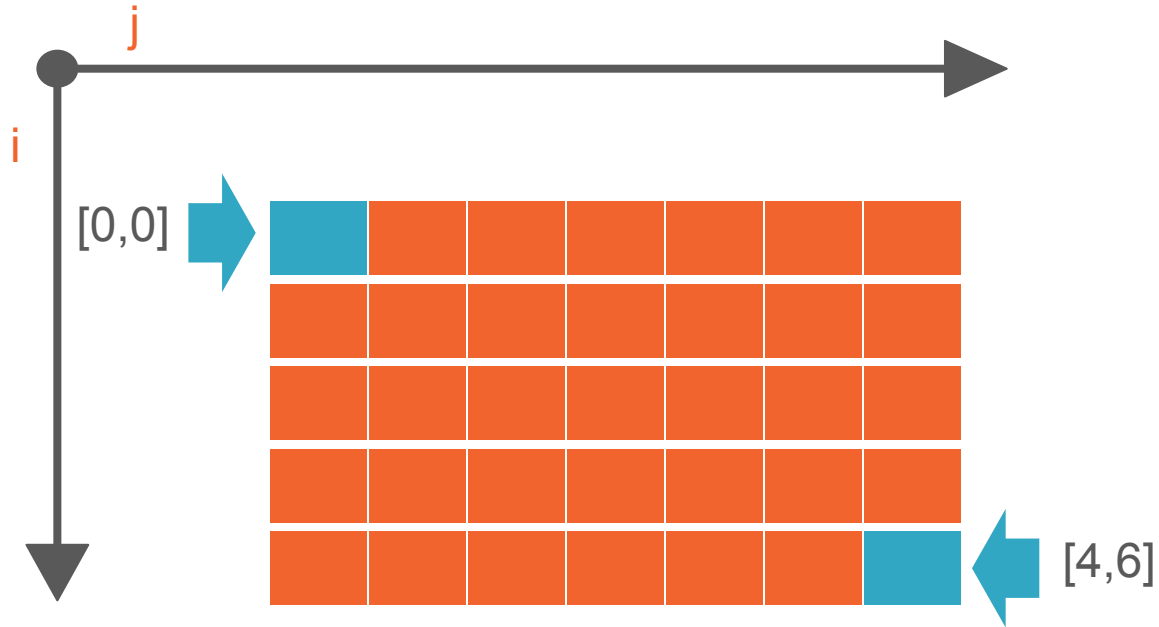
Matrices



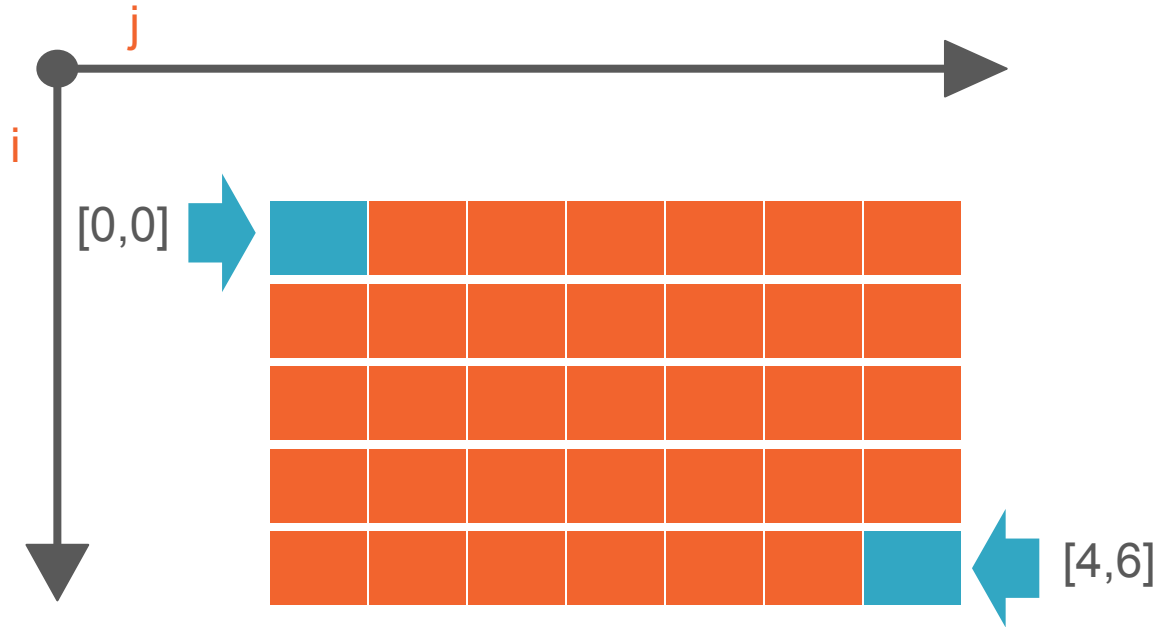
Matrices



Matrices



Matrices



$O(N^2)$

Potencia

Matrices

$O(N^2)$

Potencia

```
#define row 5
#define col 6
int tabla[row][col];

for (int i=0; i < row; i++){
    for(int j=0; j< col; j++){
        printf("celda [%i,%i] = %i \n", i, j, tabla[row][col]);
    }
}
```

Ejercicios

- Defina un vector de números enteros, solicite insertar los números por teclado dentro del vector y para finalizar muestre cual es el mayor de los números
- Define una matriz, luego permita el ingreso de datos o cree la inserción de forma automática. Para finalizar, solicite que el usuario ingrese una posición a descubrir, al revelar el valor de la celda debe identificar si es par o impar su contenido.
- NOTA: para facilitar la carga de números puede utilizar la función RAND que permite crear números random.