



ARRAYS

OPERACIONES

Oscar Paniagua @2023

Aritméticos

Operador	Significado
+	Suma dos valores
-	Resta dos valores
*	Multiplica dos valores
/	Divide dos valores
%	Devuelve el residuo de la división entera

Asignación

Operador	Significado
=	Asigna el valor de la derecha a la variable de la izquierda
+=	Suma y asigna el resultado a la variable de la izquierda
-=	Resta y asigna el resultado a la variable de la izquierda
*=	Multiplica y asigna el resultado a la variable de la izquierda
/=	Divide y asigna el resultado a la variable de la izquierda
%=	Realiza la división entera y asigna el residuo a la variable de la izquierda

Otros

Operador	Significado
&	Obtiene la dirección de memoria
sizeof	Devuelve el tamaño en bytes

Comparación

Operador	Significado (condición verdadera)
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
==	Igual a
!=	Distinto de ó no igual que

Lógicos

Operador	Significado
&&	Operador lógico AND (Y)
	Operador lógico OR (O)
!	Operador lógico NOT (NEGACIÓN)

Incremento/Decremento

Operador	Significado
++	Incrementa en 1 al valor de la variable
--	Decrementa 1 a lvalor de la variable

Operaciones con Arrays

traverse	printf	imprime todos los elementos, uno por uno
cmp	==	compara dos arrays de la misma dimensión
copy	=	copia dos arrays de la misma dimensión
sum	+	suma dos arrays de la misma dimensión

Operaciones con Arrays

(estructuras de datos)

update	modifica un elemento en la posición indicada
init rand	inicializa un array con valores aleatorios
find	busca un elemento por la posición o por el valor
insert	agrega un elemento en la posición indicada
delete	elimina un elemento en la posición indicada
delete duplicates	Elimina elementos duplicados en un array

Buscar un Elemento en un Array 1D

Buscar un valor dado, en un array de una dimensión 1D

```
int arr[5] = {1, 20, 5, 78, 30};
```

Entrada

valor = 5

Salida

Valor encontrado!

Entrada

valor = 100

Salida

Valor NO encontrado!

Buscar un Elemento en un Array 1D

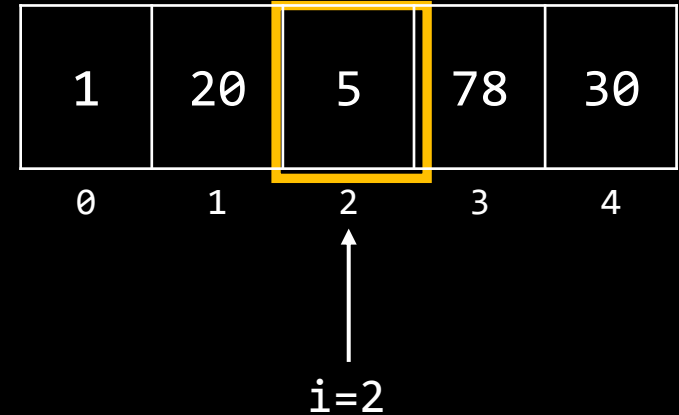
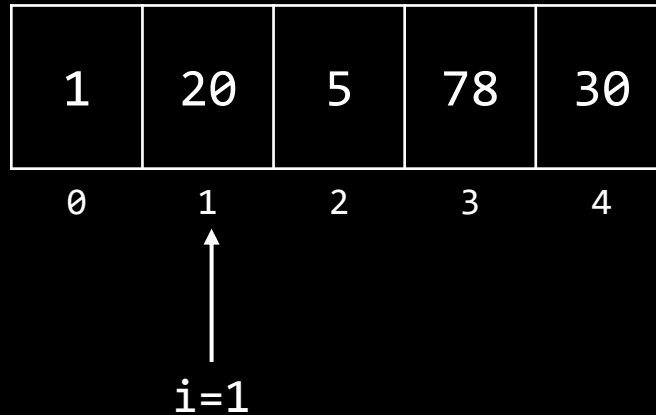
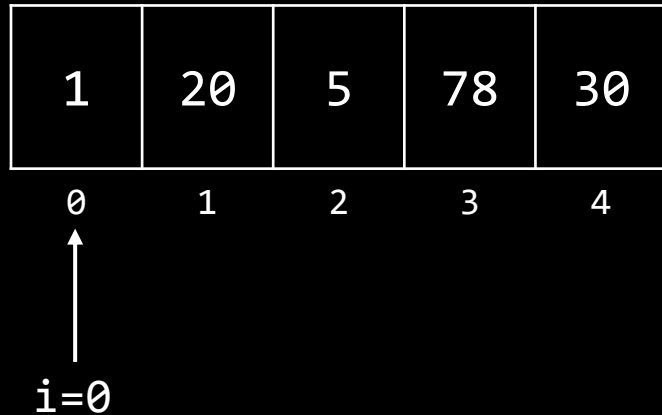
Algoritmo

- 1) Obtener el valor a buscar
- 2) Buscar el valor en el array, iterar utilizando loop (for por ejemplo)
- 3) Si se encontró, mostrar un mensaje "Encontrado"
- 4) Si no se encontró, mostrar un mensaje "No Encontrado"

Buscar un Elemento en un Array 1D

Representación Grafica

valor = 5



Operaciones con Arrays

(estructuras de datos)

update	modifica un elemento en la posición indicada
init rand	inicializa un array con valores aleatorios
find	busca un elemento por la posición o por el valor
insert	agrega un elemento en la posición indicada
delete	elimina un elemento en la posición indicada
delete duplicates	Elimina elementos duplicados en un array

Insertar un Elemento en un Array 1D

Insertar un valor dado, en una posición específica (índice) en un array de una dimensión 1D

```
int arr[10] = {10, 20, 30, 40, 50}
```

Restricciones

El índice ≥ 0 y ≤ 5 , enviar mensaje “posición invalida”

Entrada

valor = 100 posición = 2

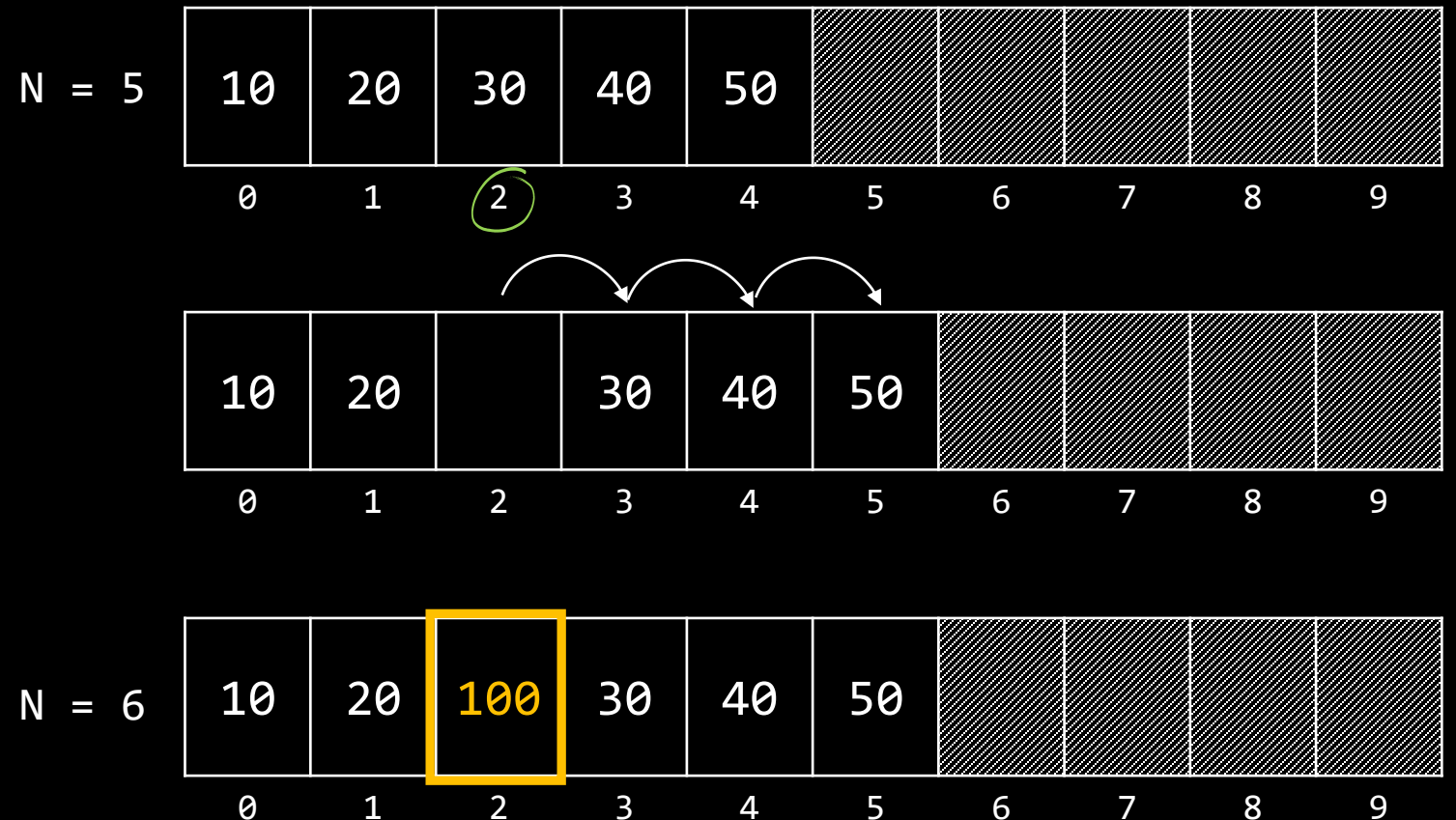
Salida

{10, 20, 100, 30, 40, 50}

Insertar un Elemento en un Array 1D

Representación Grafica

valor = 100
posición = 2
N = tamaño del array



Insertar un Elemento en un Array 1D

Algoritmo

- 1) Obtener el valor a insertar
- 2) Obtener la posición a insertar
- 3) verificar que no haya alcanzado el tamaño máximo del array
- 4) Verificar que la posición sea un valor valido (restricciones)
- 5) Si la posición es valida, desplazar todos los elementos, desde el final, una posición a la derecha, desde el lugar a insertar
- 6) Insertar el nuevo elemento
- 7) Actualizar el tamaño del array



ARRAYS

OPERACIONES

Oscar Paniagua @2023

Operaciones con Arrays

(estructuras de datos)

update	modifica un elemento en la posición indicada
init rand	inicializa un array con valores aleatorios
find	busca un elemento por la posición o por el valor
insert	agrega un elemento en la posición indicada
delete	elimina un elemento en la posición indicada
delete duplicates	Elimina elementos duplicados en un array

Remove un valor(elemento) del array

Eliminar en un array de una dimensión 1D, un valor en una posición indicada

```
int arr[5] = {1, 20, 5, 78, 30}
```

Entrada

index = 2

Salida

{1, 20, 78, 30}

Remove duplicados

Buscar valores duplicados, en un array de una dimensión 1D, y si se encuentran, removerlos

```
int arr[] = {1, 20, 5, 78, 5, 30}
```

Salida

{1, 20, 5, 78, 30}

