

Nivel \rightarrow VAR

int num = 0;

num > 0 num == 0 num == 2
 ↳ F ↳ T ↳ F

num < 0

↳ F

bool comparacion = (num > 0) $\&\&$ (num < 100)

if (num > 0)
 //

if (comparacion) $\underline{\quad}$

(num >= 0) $\&\&$ (num <= 100)
 ↓ ↓
cond 1 cond 2
 DT or of

Op. Logicas

	Caso 1	Caso 2	Caso 3	Caso 4
cond 1	T	T	F	F
cond 2	T	F	T	F
Y & &	T	F	F	F
0 > 11	T	T	T	F

switch (num) {

case 0:

 hacer();

 break;

case 1:

 hacer();

 break;

default:

 hacer();

 break;

 if (num == 0) {

 hacer();

}

 else if (num == 1) {

 hacer();

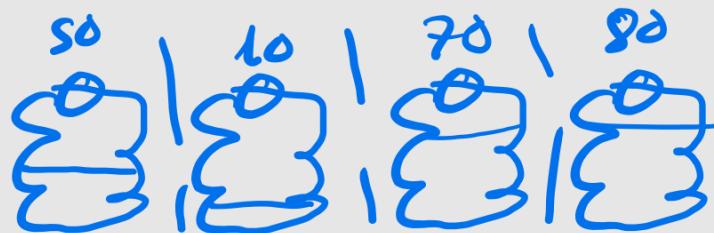
}

 else {

 hacer();

}

Bucles



> $\geq 50 \rightarrow$ Llenamos

<= 50 \rightarrow Vaciamos

int listaNiveles[4] = {50, 10, 70, 80}

WHILE (bool)

\Rightarrow Si bool == T \rightarrow REPETIR

\Rightarrow Si bool == F \rightarrow PARAR

int num = 99;

while (num > 0) {

 printf(num);
 num = num - 1; \rightarrow

{

 num = 99

 ↓
 98

 ↓
 97 $\cdots \rightarrow 0$

98 > 0

WT

0 > 0

↓
F

\rightarrow me salgo

FOR

```
for (int ind=0; ind<10; ind=F(ind+1)) {  
    printf("%d", ind);  
}
```

LISTAS → ARRAYS



> $\leq 50 \rightarrow$ Llenamos

$\leq 50 \rightarrow$ Vaciamos

```
int listaNiveles[4] = {50, 10, 70, 80}
```

```
printf("%d", listaNiveles[0])
```

```

for (int x=0 ; x <= 3 ; x=x+1) {
    printf("%d", listaNiveles[x]);
}

```

Cuál es

Bucle \Rightarrow $50 \rightarrow 0$ $x=0 \rightarrow lista[0]$
 $10 \rightarrow 1$ ↓
 $70 \rightarrow 2$ $x=1 \rightarrow lista[1]$
 $80 \rightarrow 3$ ↓

string = "Chema"
 "Angel"

char [] = "Chema"

['C', 'h', 'e', 'm', 'a', '\0']

puntero = 0x7fffd85
d1

puntero2 = 0x7fffd85

puntero3 = 0x7fffd85

n = 1000
0x7fffd85

*puntero = S0

*puntero1 = *puntero1 - 1

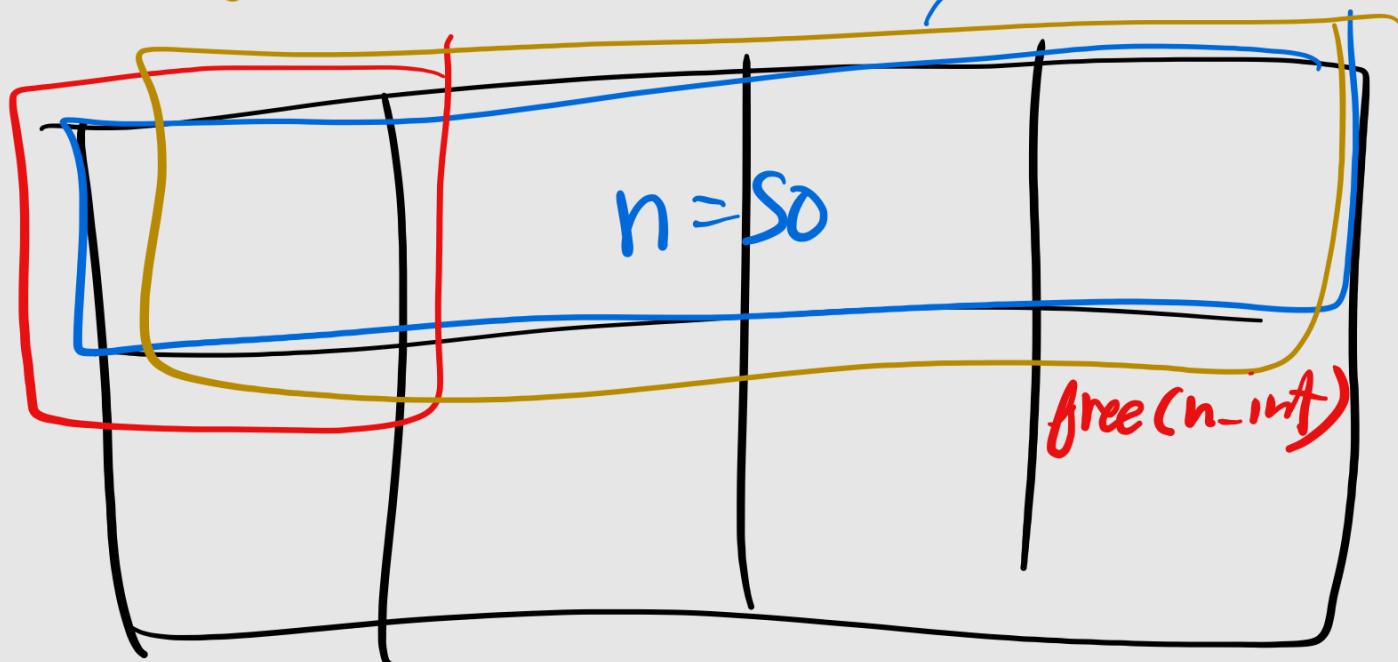
*puntero2 = *puntero2 + 2

`malloc()` $\text{int}^{\times 1} = \text{malloc}(\text{sizeof}(\text{int}))$

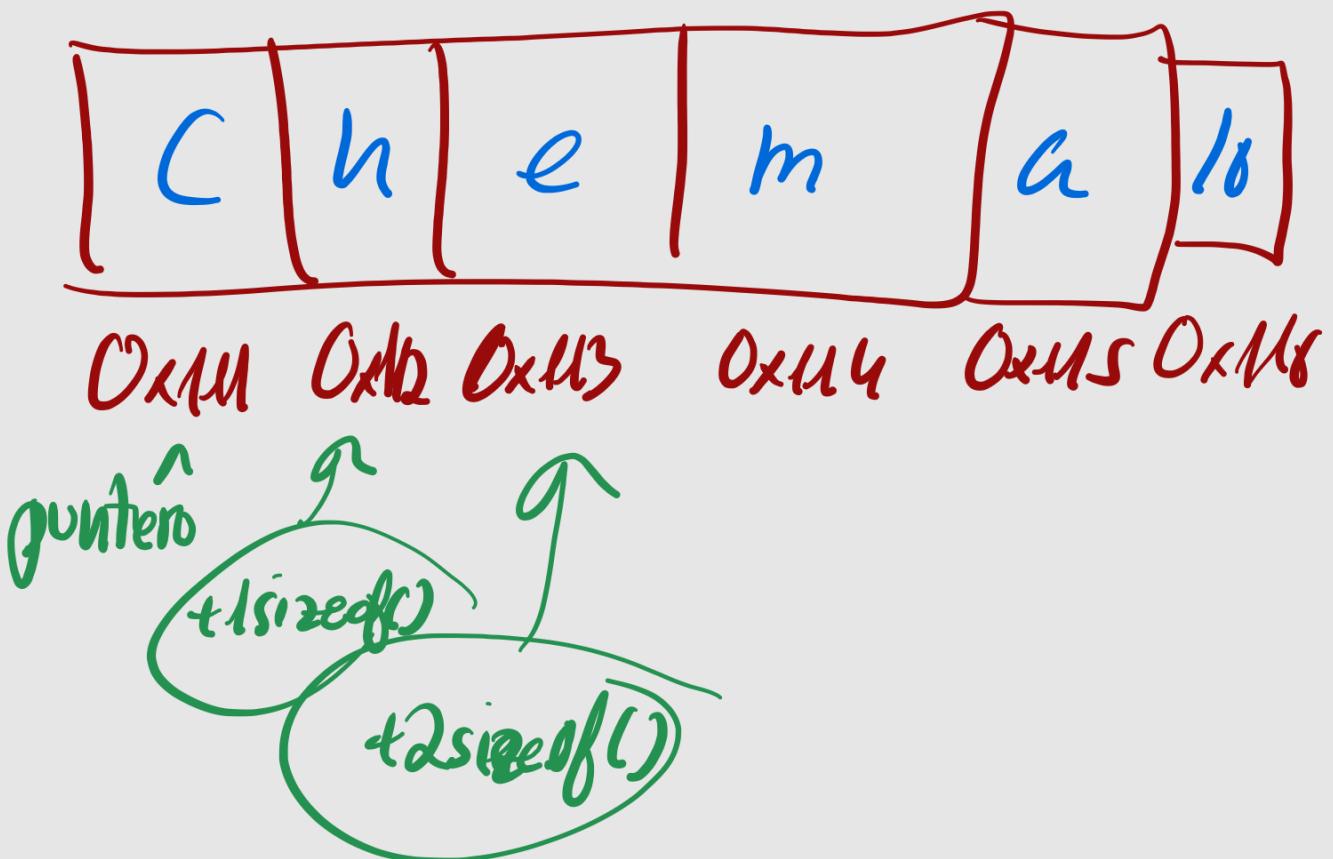
`free()`

`sizeof()` {
 $\text{ARRAY} \rightarrow \text{elems}$
 $n \text{ bytes}$
 n
 $n \text{ long bytes}$

$\text{sizeof}(\text{int})$



`char C[] = "Chelm";` $0x111 \text{ } 0x112 \dots$
imprimir (punteroNombre) \uparrow $\overset{\text{C}}{\text{h}}$ $\overset{\text{h}}{\text{0x116}}$
`void imprimirNombre (char *punteroNombre) {` $0x111$
`punteroNombre + sizeof(char) \rightarrow 0x112` h



`indice += 1;`

`[puntero + indice * sizeof(C)]`

`b0C h e m a`