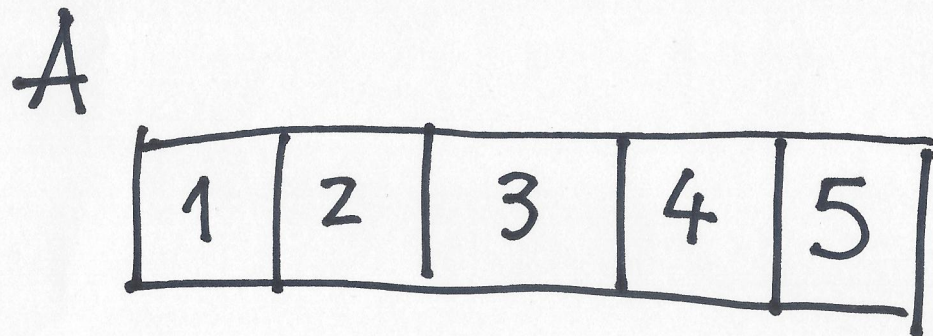


# ARREGLOS dinámicos

int A[5] = {1, 2, 3, 4, 5} → ARREGLO  
estático

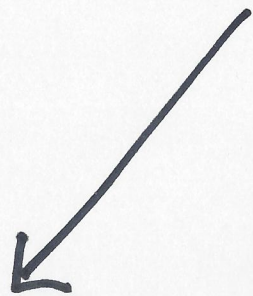


+ no es posible liberar el espacio de memoria que ocupa.

+ no es posible re-dimensionarlo.



# ARREGLOS dinámicos



Unidimensionales

Bidimensionales

+ malloc() → asignar memoria

+ calloc() → asignar e inicializar

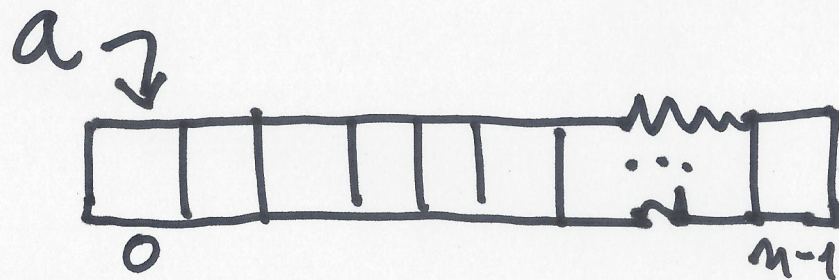
+ realloc() → redimensionar

+ free() → liberar



# ARREGLOS dinámicos (unidimensionales)

```
int *a = NULL;      a =  
    ⋮                =
```

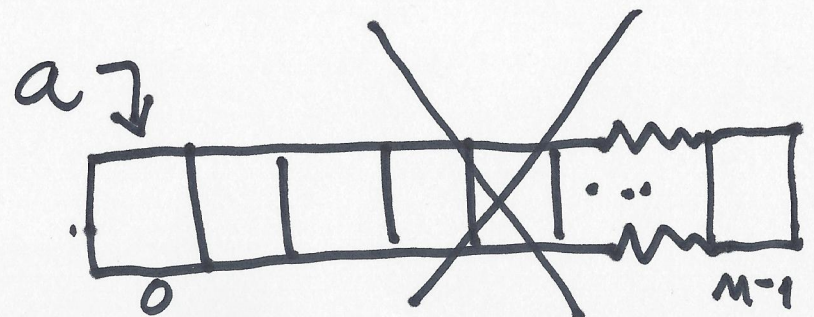


```
for (i=0; i<n; i++)
```

```
    a[i] = i+1;
```

```
    ⋮
```

```
free(a);
```

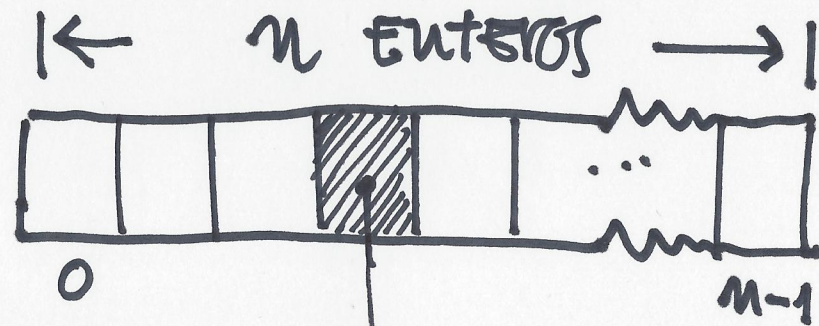




# ARREGLOS dinámicos (unidimensionales)

`int *a = NULL;`

a



cuánta memoria?  
(bytes)

sizeof(int)



sizeof(int) \*  $n$

# Elementos  
del arreglo

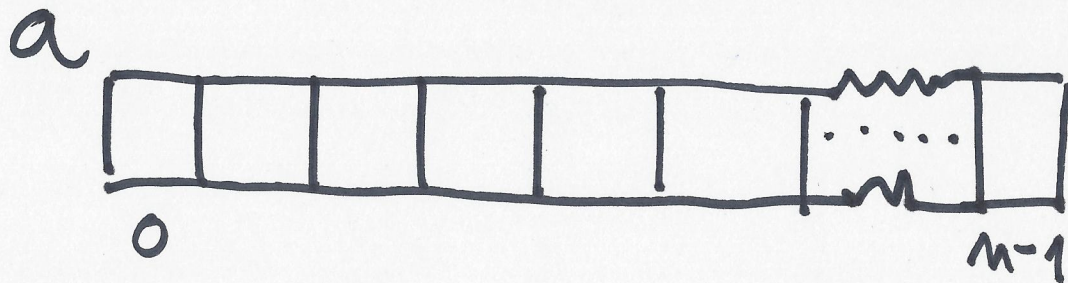


# Arreglos dinámicos (unidimensionales)

```
int *a = NULL; n;
```

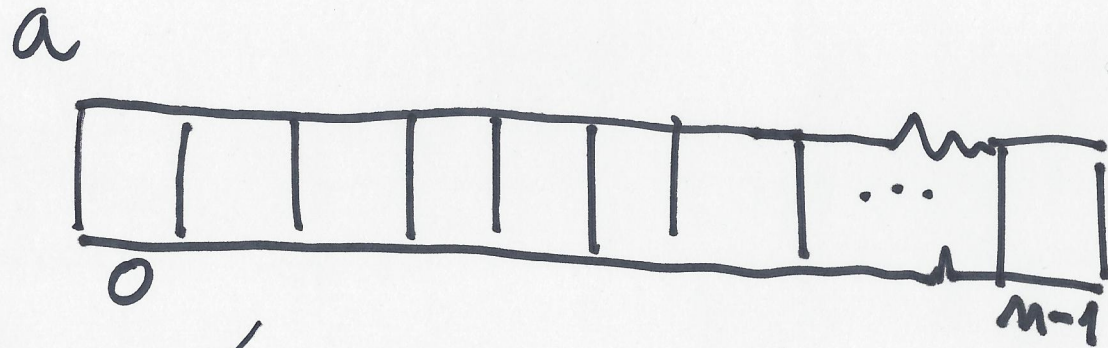
```
scanf("%d", &n);
```

```
a = (int *) malloc (sizeof(int) * n);
```

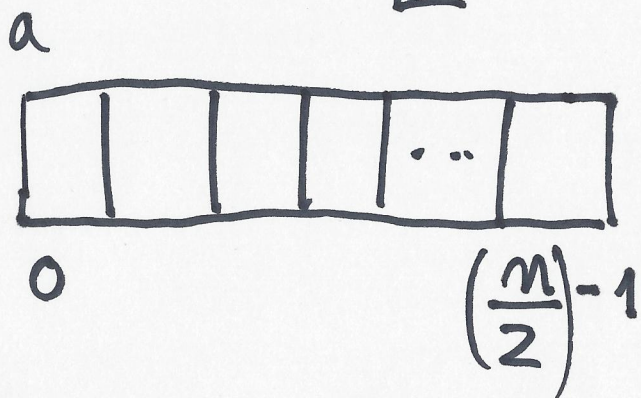




# ARREGLOS dinámicos (unidimensionales)

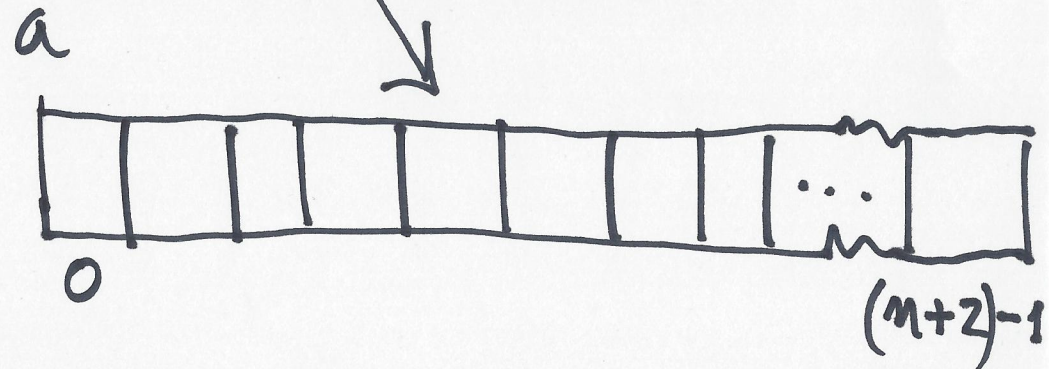


reducir a la  
mitad



realloc()

AGREGAR 2 casillas  
al arreglo

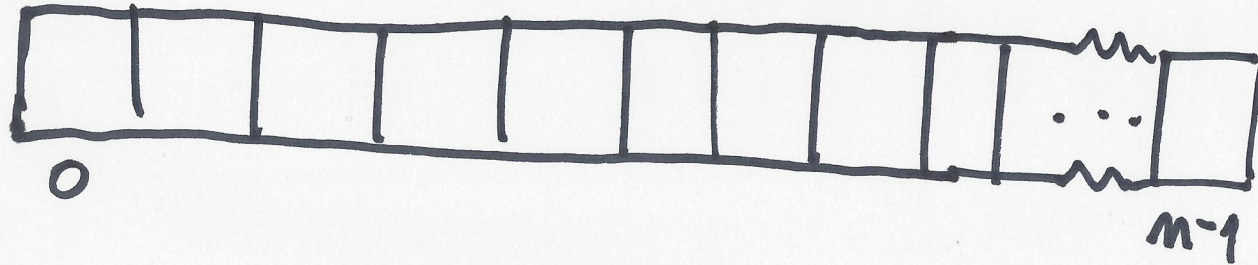




# Arreglos dinámicos (unidimensionales)

```
int *a = NULL;  
⋮
```

→ arreglo de  
n elementos



```
a = (int *) realloc(a, sizeof(int) * (n/2));
```

a

