

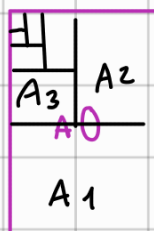
## Ejercicio 1

Input:  $i \rightarrow$  tamaño del papel

$n \rightarrow$  nro de  $\square$  a imprimir

$l \rightarrow$  largo del papel  $i$

$b \rightarrow$  ancho del papel  $i$



Caso Base  $\rightarrow$  Sería  $A_0$  ya que

a partir de él se hacen las subdivisiones

Recurción  $l/b = 2$

$i = 3$	$i = 2$	$i = 1$	$i = 0$
$l_3 = 420$	$l = b_3 \cdot 2$	$l = b_2 \cdot 2$	$l = b_1 \cdot 2$
$b_3 = 297$	$b = l_3$	$b = l_2$	$b = l_1$

## Ejercicio 2

Input: filas,  $\underbrace{\quad}_n$  columnas  $\underbrace{\quad}_m$

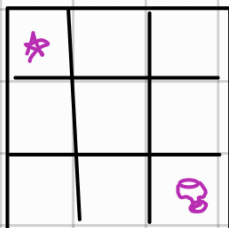
Posibles entradas

$m = 0$   
 $n = 0$  } No hay que recorrer  
                  retorna 0

$m = 1$   
 $n = 1$  } Solo hay 1 forma de  
                  llegar a la meta que es  
                  quedarse en el mismo  
                  lugar

} Casos  
Base

$m = 3$   
 $n = 3$



$m = 2$   
 $n = 3$

$m = 2$   
 $n = 2$

$m = 1$   
 $n = 3$

$m = 1$   
 $n = 2$

$m = 3$   
 $n = 2$

$m = 3$   
 $n = 1$

$= 9$  posibilidades

Calculamos las posibilidades de ir por abajo

y las sumamos con el nro de posibilidades

de ir por derecha

Recurividad

$$(m-1) + (n-1)$$

# Análisis

## Ejercicio 3

Input:  $n$  (el nro del cuál se hallará el factorial)

Se tendrá una variable global llamada

"factor" que será igual a 2, ya que ni 0 ni 1

se consideran factores primos.

### Posibles entradas:

$8 \mid 2$      $7 \mid 7$      $0 \text{ ó } 1 = \text{No existe}$   
 $4 \mid 2$      $1 \mid$     factor primo  
 $2 \mid 2$   
 $1$

### Casos Base

\*  $n=0$  ó  $n=1 \rightarrow n! = 1$

\* Cuando  $n$  es un número primo su factor primo es el mismo número.

(Se puede verificar con el  $\text{mod} == 0$ )

### Rekursividad

$n=8 \rightarrow n/\text{factor} \rightarrow 8/2 \rightarrow 4$   
 $4/2 \rightarrow 2/2 \rightarrow 1$

$n=15 \rightarrow 15/2 \rightarrow 15/3 \rightarrow 5/3 \rightarrow 5/4 \rightarrow 5/5 \rightarrow 1$   
(factor incrementará su valor en 1 hasta que sea divisible entre el nuevo valor de  $n$ )

### Pseudocódigo

```
void factorize (int n)
IF  $n == 0$  or  $n == 1$ 
return;
IF  $(n \% \text{factor} == 0)$ 
print factor
factorize ( $n/\text{factor}$ )
ELSE
++ factor
factorize ( $n$ )
```

## Ejercicio 4

Input: array, tamaño\_del\_array

Posibles entradas

size = 5

array = (1, 2, 3, 4, 5)

result = 15

size = 0

result = 0

## Pseudocódigo

```
int sumarecursiva(arr, size)
```

```
IF (size == 0)
```

```
return 0;
```

```
ELSE
```

```
return arr[size-1] + sumarecursiva(arr, size-1);
```

Caso Base size = 0 → retorna 0

(No hay nada que sumar)

Se podría decir que:

size = 5

array = (1, 2, 3, 4, 5)

5 +  
4 +  
3 +  
2 +  
1  
Recorremos una posición hasta que llegue al 1er elemento

## - Ejercicio 5

Input: base y exponente

Posibles entradas:

$3^0 = 1$      $2^2 = 4$      $0^3 = 0$

Caso base: exponente = 0  
retorna 1

## Pseudocódigo

```
int num_exp (int base, int exponente)
```

```
IF exponente == 0
```

```
return 1
```

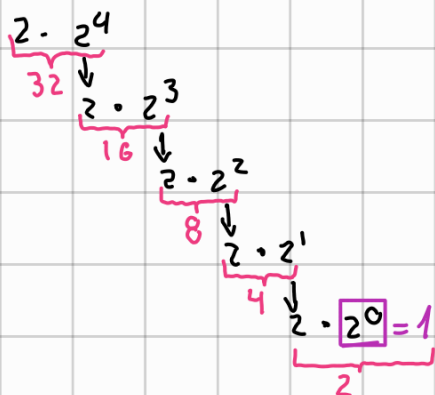
```
ELSE
```

```
return base * num_exp(base, exponente - 1);
```

Se podría decir que:

$2^5 = 2 \cdot 2^4 = 2^{1+4} = 2^5$

Recursividad



el exponente se va reduciendo en 1

## Ejercicio 6

INPUT  $N, a, b$   
 $\hookrightarrow$   $\underbrace{a, b}_{\text{coordenadas}}$   
 $\hookrightarrow$  Tamaño tablero

$$a \geq 1 \quad x \leq N$$

$$b \geq 1 \quad b \leq N$$

$$\text{tablero}(a, b) = 0$$

$$\text{tablero}(a, b) = ?$$

Recurividad  
 $(a, b, i+1)$

## - Ejercicio 7

Input:  $n$  (enésimo número a calcular)

Posibles entradas

$n=0$  (en nù caso sería 0)

$n=1$  (retorna 1)

$n=6$  (retorna 8)

$\left[ \overset{1}{1}, \overset{2}{1}, \overset{3}{2}, \overset{4}{3}, \overset{5}{5}, \overset{6}{8}, \overset{7}{13}, \overset{8}{21}, \overset{9}{34}, \overset{10}{55} \right]$

Casos base ①  $n=0 \rightarrow 0$   
 ②  $n=1 \rightarrow 1$

Se podría decir que:

$$n=4 \rightarrow (n-1) + (n-2) = 3 + 2 = 2 + 1 = 3$$

Posición  $\swarrow$  Valor de la posición

$$\begin{array}{ccccccc}
 & & n=4 \rightarrow 3 & & & & \\
 & 2 & & 3 & + & 2 & 1 \\
 1+1 & \underbrace{(3-1)+(3-2)} & + & \underbrace{(2-1)+(2-2)} & 1+0 \\
 1+0 & \underbrace{(2-1)+(2-2)} & & & & & \\
 \text{Caso Base} & \text{Caso Base} & & \text{Caso Base} & & & 
 \end{array}$$

La recursividad va en sumar los dos valores anteriores a la posición  $n$

Pseudocódigo

int fibonacci (int n)

IF  $n == 0$

return 0;

IF  $n == 1$

return 1;

ELSE

return fibonacci(n-1) + fibonacci(n-2)