

# Prog. Básica - Laboratorio 6

## Subprogramas

Nombre: \_\_\_\_\_ Fecha: \_\_\_\_\_

### NOTA:

- Los casos de prueba dados no son suficientes! Debéis probar mas casos.
- Nosotros usaremos nuestros propios casos de prueba para testear.
- A la hora de entregar los programas, copiad todos los programas dentro de una carpeta, darle vuestro nombre y apellido, comprimirlo y entregar.
- **IMPORTANTE:** en los programas de los casos de prueba, fijaros si un subprograma ha de ser procedure o function, solo hay una única forma y **NO SE PUEDE MODIFICAR**, como **TAMPOCO SE PUEDE MODIFICAR** el nombre del subprograma.

### 1. Parchís

Escribir un programa para sacar una ficha del Parchís de su casa. Una ficha saldrá de su casa si lanzando un dado se obtiene un cinco. De no obtener un cinco se podrá lanzar el dado sucesivas veces mientras no se superen cuatro intentos. Para desarrollar el algoritmo se podrán invocar los siguientes subprogramas: *sacar\_ficha*, *lanzar\_dado*.

#### 1) Especificación

Salida: mensaje indicando si la ficha sale o no

Post: una ficha saldrá de casa si se obtiene un 5 con el dado en alguno de los 4 posibles intentos.

Superados los intentos imprimir mensaje: “Límite de intentos superado”

Especificación de *sacar\_ficha*:

Salida: muestra por pantalla el mensaje: “Ficha sale”

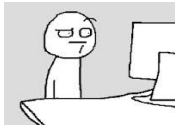
Especificación de *lanzar\_dado*:

Resultado: numero:  $e_i$

Post:  $1 \leq e_i \leq 6$

### Plantillas ofrecidas:

sacar_ficha.adb	Para hacer el programa en Ada
lanzar_dado.adb	Para hacer el programa en Ada
prueba_parchis.adb	Para probar el programa en Ada



## 2. Multiplicación por sumas

La tecla “\*” esta atascada y por lo tanto inutilizable. Diseña un algoritmo que dados dos números calcule el resultado de multiplicar ambos.

### 1) Especificación

Datos: 2 números enteros: n y multiplicador

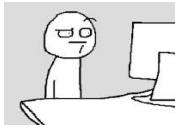
Pre:  $0 < n$ , multiplicador  $\leq 10$

Resultado: Un número entero: resultado

Post: resultado =  $n * \text{multiplicador}$

### Plantillas ofrecidas:

multporsumas.adb	<i>Para hacer el programa en Ada</i>
prueba_multporsumas.adb	<i>Para probar el programa en Ada</i>



### 3. División por restas

Ahora la tecla “/” también está atascada. Crea un algoritmo que dados un número entero que llamaremos dividendo ( $0 \leq \text{dividendo} \leq 100$ ) y otro número entero que llamaremos divisor ( $0 < \text{divisor} \leq 10$ ), calcule el resultado de la división de ambos números y el resto, sin utilizar las operaciones / ni rem: es decir, realizando la división como una repetición de restas sucesivas. Por ejemplo  $6/2 = 6-2=4-2=2-2=0$ , como hemos realizado 3 restas significa que el resultado es 3, y el resto es 0.  $7/2=7-2=5-2=3-2=1$ , como hemos realizado 3 restas el resultado es 3 y el resto es 1.

#### 1) Especificación

Datos: 2 números enteros: dividendo y divisor

Pre:  $0 \leq \text{dividendo} \leq 100$  y  $0 < \text{divisor} \leq 10$

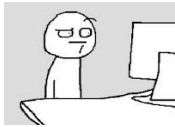
Resultado: 2 números enteros: cociente y resto

Post: cociente  $\geq 0$ . cociente contiene el valor del cociente entero de dividendo/divisor

$0 \leq \text{Resto} < \text{divisor}$ . Resto contiene el valor del resto de dividir dividendo/divisor  $< \text{divisor}$

#### Plantillas ofrecidas:

diviporrestas.adb	<i>Para hacer el programa en Ada</i>
prueba_diviporrestas.adb	<i>Para probar el programa en Ada</i>



#### 4. Múltiplos de 3 entre dos enteros

Construye un algoritmo que, dados dos números naturales,  $N1$  y  $N2$ , tales que  $N1$  es estrictamente menor que  $N2$ , escriba por pantalla cada uno de esos números y devuelva la suma de los múltiplos de 3 entre  $N1$  y  $N2$  (ambos inclusive).

Por ejemplo, si  $N1=5$  y  $N2=15$  se debe escribir “ 6 9 12 15” y devolver que la suma es 42 (es decir,  $42=6+9+12+15$ ).

##### 1) Especificación

Datos: 2 números enteros:  $n1$  y  $n2$

Pre:  $0 < n1, n2$

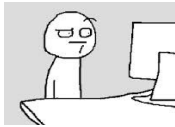
Resultado: Un número entero: suma

Post: suma = números entre  $n1$  y  $n2$  múltiplos de 3

Salida: Secuencia de enteros que tiene los números entre  $n1$  y  $n2$  que son múltiplos de 3

##### Plantillas ofrecidas:

multiplosdetres.adb	<i>Para hacer el programa en Ada</i>
prueba_multiplosdetres.adb	<i>Para probar el programa en Ada</i>



## 5. Divisores

¡Por fin! Todas las teclas funcionan de nuevo. Ahora podemos utilizar todas las operaciones. Diseñar un algoritmo donde dado un numero entero, calcule y escriba por pantalla los divisores de un número n.

### 1) Especificación

Datos: 1 número entero

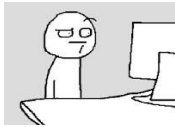
Pre:  $0 < \text{num:valor1} \leq 20$

Salida: dos o más números enteros divisores\_de\_num

Post:  $1 \leq \text{divisores\_de\_num} \leq n \mid \text{divisores\_de\_num} \in [1..n]$

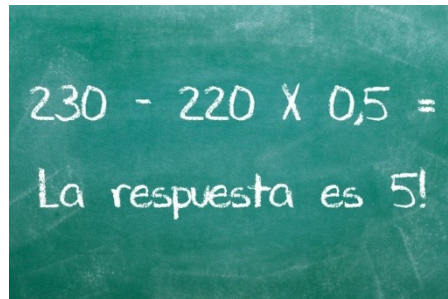
### Plantillas ofrecidas:

divisores.adb	<i>Para hacer el programa en Ada</i>
prueba_divisores.adb	<i>Para probar el programa en Ada</i>



## 6. Factorial

Crea un algoritmo para calcular el factorial de un número entero  $n < 6$  dado



[https://verne.elpais.com/verne/2019/07/15/articulo/1563188505\\_109036.html](https://verne.elpais.com/verne/2019/07/15/articulo/1563188505_109036.html)

### 1) Especificación

Datos: 1 número entero

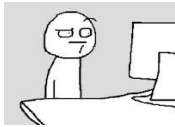
Pre:  $0 \leq \text{num} \leq 6$

Resultado: un número entero: factorial

Post:  $0 < \text{factorial} \leq 720 \mid \text{factorial} = \prod_{i=1}^{\text{num}} i$ , siendo  $0! = 1$

### Plantillas ofrecidas:

factorial.adb	<i>Para hacer el programa en Ada</i>
prueba_factorial.adb	<i>Para probar el programa en Ada</i>



## 7. Número triangular

Crea un algoritmo para calcular el número triangular de un número dado del 1 al 10 (inclusive).

### 1) Especificación

Datos: 1 número entero

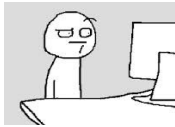
Pre:  $0 < \text{num} \leq 10$

Resultado: un número entero: num\_triangular

Post:  $0 < \text{num\_triangular} \leq 55 \mid \text{num\_triangular} = \sum_{i=1}^{\text{num}} i$

### Plantillas ofrecidas:

numtriangular.adb	<i>Para hacer el programa en Ada</i>
prueba_numtriangular.adb	<i>Para probar el programa en Ada</i>



## 8. Dibujar gráfico

Crea un algoritmo que dado un número positivo  $n$  y, según su valor, dibuje un gráfico de  $n$  líneas como el de la figura.

### 1) Especificación

Datos: 1 número entero

Pre:  $0 < \text{num} \leq 10$

Salida: un dibujo por pantalla. Por ejemplo, para  $\text{num}=5$

```
*0000
**000
***00
****0
```

para  $\text{num}=3$

```
*00
**0
***
```

### Plantillas ofrecidas:

grafico.adb	<i>Para hacer el programa en Ada</i>
prueba_grafico.adb	<i>Para probar el programa en Ada</i>