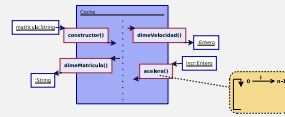


Grado Tecnologías Interactivas

Programación 1



Práctica 2

UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escola Politècnica Superior de Gandia

DSIC

Departament de Sistemes Informàtics i Computació

Práctica 2

Objetivos

- Implementación de algoritmos sencillos mediante sentencias condicionales (`if`) y bucles (`for` o `while`).
- Implementación del diseño de funciones sencillas.

¡ Atención !

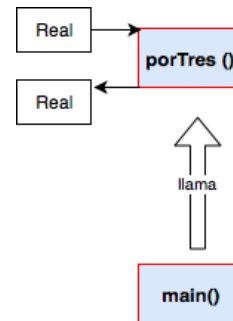
- ▷ Se recuerda que las prácticas deben prepararse antes de acudir al aula informática, anotando en el enunciado las dudas que se tengan.
- ▷ Los diseños y algoritmos que se piden en esta práctica deben escribirse en la libreta de apuntes para poder ser revisados.
- ▷ La realización de las prácticas es un trabajo individual y original. En caso de plagio se excluirá al alumno de la asignatura. Por tanto, es preferible presentar el trabajo realizado por uno mismo aunque éste tenga errores.



1

Funciones sencillas

1. La función `porTres()` recibe un número real y devuelve el resultado de multiplicarlo por tres. Su diseño es el siguiente:



Y su algoritmo es éste:

Datos de entrada: $a : \mathbb{R}$

Datos de salida: \mathbb{R}

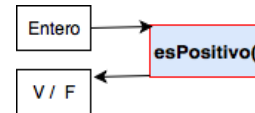
Devolver $a \cdot 3$

Escribe un programa `PorTres.cpp` en el que se llame a la función `porTres()` desde `main()`.

- ▷ Atención: "devolver" no es escribir en pantalla.
- ▷ Atención: Hay dos tipos mutuamente excluyentes de funciones:
 - las de entrada/salida: se comunican con el usuario (escriben en pantalla y/o leen de teclado). No hacen cálculos. Ejemplo: `main()`.
 - las de cálculo: no se comunican con el usuario (ni leen ni escriben), sólo calculan. Ejemplo: `porTres()`.



2. La función `esPositivo()` recibe un número entero y devuelve verdadero si es mayor o igual que cero; o falso en caso contrario. Su diseño es el siguiente:



Y su algoritmo es éste:

```
Datos de entrada:  $a : \mathbb{Z}$   
Datos de salida:  $\mathbb{B}$  (Boolean)
```

```
si  $a \geq 0$   
|  
| Devolver verdadero  
|  
|  
si no  
|  
| Devolver falso  
|
```

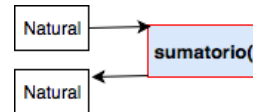
Escribe y prueba la función `esPositivo()` en un programa llamado `EsPositivo.cpp`.



2

Funciones con bucles

1. La función `sumatorio()` recibe un número natural n y devuelve la suma $1+2+3+\dots+n$. Es decir, $\sum_1^n i$. Su diseño es:



Un algoritmo para esta función sin bucles es el siguiente. Véase que una fórmula matemática para calcular algo es también un algoritmo:

Datos de entrada: $n : \mathbb{N}$

Datos de salida: \mathbb{N}

Devolver $\frac{n \cdot (n+1)}{2}$

Utilizando una repetición, podemos implementar `sumatorio()` con este otro algoritmo.

Datos de entrada: $n : \mathbb{N}$

Datos de salida: \mathbb{N}

$res \leftarrow 0$

1 \xrightarrow{i} n

$res \leftarrow res + i$

Devolver res

Escribe un programa `Sumatorio.cpp` con ambas versiones de la función `sumatorio()`.



2. Diseña una función `sumatorioPares()` que reciba un número natural n y devuelva la suma de los números naturales pares menores o iguales que n .

Un algoritmo para esta función podría ser:



Piensa un algoritmo para la función `sumatorioPares()` que utilice una repetición pero no una condición dentro de ella.

¿Se te ocurre alguna fórmula que calcule directamente la suma de los números pares menores o igual que n ?

Escribe un programa `SumatorioPares.cpp` con dos versiones diferentes de la función `sumatorioPares()`.



3. Diseña una función `esPrimo()` que reciba un numero natural y devuelva verdadero si éste es un número primo. Un número natural es primo si es divisible únicamente por uno y por sí mismo.
Escribe un algoritmo sencillo para determinar si un número n es o no primo.
Escribe un programa `EsPrimo.cpp` para comprobar tu implementación de la función `esPrimo()`.



13 septiembre 2018