



## Práctica N° 5 Recursividad

### Ejercicios de clase.

1. Implementa una función recursiva que calcule la suma de los  $n$  primeros números naturales. La cabecera sería la siguiente:

```
unsigned sumaNaturales(unsigned n)
```

donde el parámetro  $n$  es el número de naturales a sumar. Diseña la función principal (main) para probar el funcionamiento de `sumaNaturales`.

2. El valor de la función potencia  $x^n$ , se puede definir recursivamente del modo siguiente:

$$x^n = 1 \text{ si } n=0$$

$$x^n = x * x^{n-1} \text{ si } n \geq 1$$

Implementa una función potencia que calcule recursivamente el valor de  $x^n$  con la siguiente cabecera:

```
unsigned potencia(unsigned x, unsigned n)
```

Diseña la función principal (main) para probar el funcionamiento de `potencia`.

3. Implementa una función recursiva que calcule el producto de dos número naturales  $x$  e  $y$ . La cabecera sería la siguiente:

```
unsigned producto(unsigned x, unsigned y)
```

A la hora de diseñar la solución ten en cuenta que los únicos operadores aritméticos que puedes usar son la suma y la resta. Diseña la función principal (main) para probar el funcionamiento de `producto`.

4. Implementa un procedimiento recursivo que imprima los dígitos de un número natural  $n$  en orden inverso. Por ejemplo, para  $n=675$  la salida debería ser 576. La cabecera sería la siguiente:

```
void inverso (unsigned n)
```

Diseña la función principal (main) para probar el funcionamiento de `inverso`.

## Ejercicio de refuerzo.

5. Implementa una función recursiva que devuelva true si el número que se le pasa como parámetro es primo y false en caso contrario. La cabecera de la función sería la siguiente:

```
bool esPrimoRec (unsigned num, unsigned divisor)
```

en el primer parámetro le habríamos de pasar el número, y el segundo parámetro es un número para el que hay que comprobar si es o no divisor. Inicialmente (en la primera llamada a `esPrimoRec`) el valor de ese parámetro es 2.

Diseña la función principal (main) para probar el funcionamiento de `esPrimoRec`.

6. Implementa un procedimiento recursivo al que se le pase como parámetro un número n en base 10 (decimal), e imprima su valor en base 2 (binario). La cabecera sería la siguiente:

```
void decimalAbinario (unsigned n)
```

Por ejemplo para `n=23`, el procedimiento debería imprimir 10111.

Diseña la función principal (main) para probar el funcionamiento de `decimalAbinario`.