

Ejercicio 1

Diseñe una función recursiva en C que permita transformar un número n , escrito en base b (siendo $b \leq 9$), a base 10.

Ejemplo: si $n = 256$ en base $b = 7$, n en base 10 será $= 6 \cdot 7^0 + 5 \cdot 7^1 + 2 \cdot 7^2 = 6 + 5 \cdot (7) + 2 \cdot (7 \cdot 7) = 139$.

El prototipo de la función podría ser:

int sumaDigitos(int n, int b), donde n es el número a transformar, y b es la base.

SE PIDE:

1. Diseñar la correspondiente función recursiva (puede ser recursiva no final o final), y la sentencia que debe llamar a dicha función para obtener el resultado cuando $n = 256$ y $b = 7$.
2. ¿Cuál es el tamaño del problema?
3. ¿Calcular $T(n)$ para el caso mejor y peor de la función recursiva propuesta?

AYUDA: Como se ha supuesto $b \leq 9$, para obtener el último dígito de un número entero en el lenguaje C, puede utilizar la operación módulo, $n \% 10$. Para obtener todos los dígitos de un número menos último, puede dividir el número por 10 (división entera).

SOLUCIÓN

Recursivo Lineal No final

```
int bTo10(int n, int b) {                                bTo10(256, 7)
    int r;
    if (n < 10) {
        r = n;
    } else {
        r = n%10 + b * bTo10(n/10, b);
    }
    return r;
}
```

```
int bTo10(int n, int b, int acum) {                      bTo10(256, 7, 1)
    int r;
    if (n < 10) {
        r = n*acum;
    } else {
        r = (n%10)*acum + bTo10(n/10, b, acum * b);
    }
    return r;
}
```

Recursivo Final

```
int bTo10Final(int n, int b, int acum, int sol) {        bTo10Final(256, 7, 1, 0)
    int r;
    if (n < 10) {
        r = sol + n*acum;
    } else {
        r = bTo10Final(n/10, b, acum * b, sol + (n%10)*acum);
    }
    return r;
}
```

Tamaño n = número de cifras que tiene el número.

$T(n) = T(n-1) + k$ de acuerdo con la fórmula si $a = 1$ entonces $T(n) = c_1 n + c_2$. Orden lineal si c_1 distinto de cero.

Si n es el número en si, $T(n) = T(n/10) + k$, y el orden sería logarítmico (base 10)