

Ejercicio 1

Dado un número n , se desea determinar el número de divisores de dicho número distintos de 1 y del propio n (1 y n siempre van a ser divisores de n , y por lo tanto, no se van a tener en cuenta). La solución a dicho problema se puede definir recursivamente de la siguiente forma:

$$\text{divisores}(n, k) = \begin{cases} 0 & \text{si } k = n \\ \text{divisores}(n, k + 1) & \text{si } (k < n) \text{ Y } (n \% k \neq 0) \\ 1 + \text{divisores}(n, k + 1) & \text{si } (k < n) \text{ Y } (n \% k = 0) \end{cases}$$

siendo la llamada inicial $\text{divisores}(n, 2)$.

SE PIDE:

1. Implemente en lenguaje C el método `int divisores(int n, int k)` acorde a la definición recursiva proporcionada.
2. Indique cuál es el tamaño y el orden de complejidad del método implementado en el apartado anterior. Justifique su respuesta.
3. Para dicho problema:
 - a. Proporcione una definición **recursiva final**.
 - b. Indique cuál sería la llamada inicial para dicha definición recursiva final.
 - c. Implemente en lenguaje C el método `int divisoresFinal(int n, int k, int ac)` acorde a dicha definición recursiva final.
4. Implemente en lenguaje C el método `int divisoresIterativo(int n, int k)` que resuelva el problema de forma **iterativa**.

Puntuación: 3 puntos

Solución

```
1. int divisores(int n, int k){
    int res;
    if(k == n)
        res = 0;
    else if(n%k == 0)
        res = 1 + divisores(n,k+1);
    else
        res = divisores(n,k+1);
    return res;
}
```

2. Tamaño: $n' = n - k$

Orden de complejidad:

$$T(n') = T(n'-1) + c$$

$$A = 1, b = 1, c = 1, d = 0 \rightarrow \theta(n')$$

3.a)

$$\text{divisoresFinal}(n, k, ac) = \begin{cases} ac & \text{si } k = n \\ \text{divisoresFinal}(n, k + 1, ac) & \text{si } (k < n) \text{ Y } (n \% k \neq 0) \\ \text{divisoresFinal}(n, k + 1, ac + 1) & \text{si } (k < n) \text{ Y } (n \% k = 0) \end{cases}$$

3. b) Llamada inicial `divisoresFinal(n, 2, 0)`.

```
3. c) int divisoresFinal(int n, int k, int ac){
    int res;
    if(k == n)
        res = ac;
    else if(n%k == 0)
        res = divisoresFinal(n, k+1, ac+1);
    else
        res = divisoresFinal(n, k+1, ac);
    return res;
}
```

```
4. int divisoresIterativo(int n, int k){
    int res;
    int ac = 0;

    while(k != n){
        if(n%k == 0)
            ac++;
        k++;
    }

    res = ac;
    return res;
}
```