

Diseño de Algoritmos Recursivos

22 de marzo de 2017

1 Laboratorio

1. La n -ésima potencia de 2, se define recursivamente como sigue:

$$2^n = \begin{cases} 1 & \text{si } n = 0 \\ 2 \cdot 2^{(n-1)} & \text{si } n > 0 \end{cases}$$

Escribir el código Ada de una función

function Potencia (N : Natural) return Natural;

que calcula recursivamente la n -ésima potencia de 2.

2. El inverso de un string es otro string con los mismos caracteres pero en el orden inverso. Definir recursivamente una función que dado un string calcule su inverso. Escribir el código Ada de un subprograma

function Dar_La_Vuelta(S : String) return String;

que devuelva el string inverso de S.

3. Definir recursivamente una función que dado un string decida si aparece el carácter 'a'. Escribir el código ADA de un subprograma

function Contiene_a (S : String) return Boolean;

que devuelve el valor *true* cuando S contenga una 'a'.

4. Definir recursivamente una función que dado un string devuelva el número de apariciones del carácter 'a'. Escribir el código ADA de un subprograma

function Num_apariciones_a (S : String) return Natural;

que devuelve el número de veces que aparece 'a' en S.

5. Definir recursivamente una función que dado un string decida si contiene el string "aa" como substring. Escribir el código ADA de un subprograma

function Substring_aa (S : String) return Boolean;

que devuelve el *true* si el string "aa" aparece como substring de S.