Implementación de funciones básicas recursivas usando el lenguaje LISP

Implementation of basic recursive functions using LISP.

Autor: Jhonatan Ospina Osorio
Ingeniería de sistemas, Universidad tecnológica de Pereira, Pereira, Colombia
Correo-e: Jhonatan.ospina@utp.edu.co

***Resumen*— En este documento se evidenciará mi perspectiva con respecto a LISP y veremos tres ejemplos recursivos usando el programa compilador.**

***Palabras clave*— Inteligencia, Artificial, LISP.**

***Abstract*— in this document will be show the perspective I got of the software LISP and will see three recursive examples using de compiler.**

***Key Word* — Intelligence, Artificial, LISP.**

I. INTRODUCCIÓN

Por lo que pude ver en el primer documento de LISP se evidencia un arduo trabajo en la explicación básica para su manejo, su nacimiento y por qué es tan querida en el campo de la inteligencia artificial.

II. CONTENIDO

LISP me recordó a uno de los leguajes de programación que vimos al principio de la carrera, Scheme, ya que su principal característica es la programación prefija y que se deriva de uno de los lenguajes gramaticales y matemáticos famosos por este método, el lenguaje lambda (λ).
También vamos a ver la implementación de tres funciones en este leguaje, los cuales son:

- Calcular la sumatoria de los números hasta el valor N.
- Calcular e imprimir la factorial del número X.
- Calcular e imprimir el valor absoluto del número X.

Para poder aplicar estas funciones y darnos cuenta si funcionan procedemos a usar el software online que compila y ejecuta el código en lenguaje LISP y nos permite una mejor visualización de lo que estamos programando.
A continuación, con cada uno de los ejemplos se adjuntará una imagen con su código en la plataforma y el resultado que este nos debe devolver.

Primera implementación:
Para calcular la sumatoria usaremos el método recursivo muy bien conocido.

```
(defun sumatoria (n)
    (cond
        (
            (= n 0) 0)
            (t (+ n (sumatoria(- n 1)))
        )
    )
)
(write (sumatoria 5))
```

El resultado obtenido con este ejemplo es quince, y es verificado en la plataforma.

Segunda implementación:
Para calcular la factorial de un numero usaremos la recursividad y además debemos de usar unas condiciones que nos permitan separar los casos especiales.

```
(defun factorial (n)
    (cond
        ((= n 0) 1)
        ((= n 1) 1)
        (t (* n (factorial(- n 1))))
    )
)
(write (factorial 5))
```

El resultado obtenido con este ejemplo es ciento veinte, lo cual es correcto a la hora de verificar con un recurso externo.
Notemos que el resultado de los ejemplos es específicamente del caso que se muestra en la imagen adjuntada para cada caso, estos ejemplos se pueden evaluar con cualquier número y en ese caso deben funcionar.

Fecha de Recepción: (Letra Times New Roman de 8 puntos)
Fecha de Aceptación: Dejar en blanco

Tercera implementación:
Para calcular el valor absoluto de un número es tan fácil como verificar si el número es negativo y multiplicarlo por menos uno.

```
(defun absoluto (n)
    (cond
        ((< n 0) (* n -1))
        (t n)
    )
)
(write (absoluto -5))
```

El resultado obtenido de este ejemplo fue cinco, lo cual es correcto, además, si se una un numero positivo n, el código regresara el mismo valor, ya que el valor absoluto de un positivo es el mismo.

III. CONCLUSIONES

La programación de LISP parece más una combinación entre lo que es lambda con Python básico orientado al manejo de datos y agrupación de los mismos.

IV. REFERENCIAS

Sitio web para compilar código LISP:
https://www.tutorialspoint.com/execute_lisp_online.php