

BOLETÍN PRÁCTICA 7 Programación en LISP

OBJETIVOS



Tras la realización de este boletín, y del estudio de la parte correspondiente de la asignatura, deberás ser capaz de:

- Implementar funciones en LISP.

TEMÁTICA

Tras haber estudiado el lenguaje LISP, en esta práctica se trata de practicar la programación en el lenguaje mediante la realización de programas de mayor dificultad a los realizados en el aula.

TRABAJO PREVIO

- Diseño de las estrategias que se utilizarán para resolver el problema.

La sesión de prácticas se dedicará a implementar el enfoque, y no a diseñar las estrategias necesarias para la resolución del ejercicio.

ENUNCIADO

1. Definir una función llamada **subconjuntos** que a partir de una lista genere todos los posibles subconjuntos que puedan formarse con sus elementos. Por ejemplo:

```
> (subconjuntos '(a b))  
((a b) (a) (b) ())  
> (subconjuntos '(1 2 3))  
((1 2 3) (1 2) (1 3) (1) (2 3) (2) (3) NIL)
```

Contempla dividir la solución en varias funciones

2. Una forma de representar matrices es como una lista de listas, donde cada fila viene representada por una lista de los elementos que la forman. Implementar una función llamada **transpuesta** que dada una matriz M expresada en este formato devuelva su transpuesta. Como en el caso anterior, contempla el uso de varias funciones.

Ejemplos:

```
> (transpuesta '((1 2 3 4) (5 6 7 8))  
((1 5) (2 6) (3 7) (4 8))  
> (transpuesta (transpuesta '((1 2 3) (4 5 6))))  
((1 2 3) (4 5 6))
```

3. Implementa una función llamada **maximo** que calcule el máximo de una lista anidada de números

Ejemplos:

```
> (maximo '())
```

```
NIL
```

```
> (maximo '(-3 4 (2 8)))
```

```
8
```

```
> (maximo '(-2 -4 -6))
```

```
-2
```

4. Implementa una función llamada **aplicarFunciones** que reciba como parámetros una lista y dos funciones unarias, y devuelva el resultado de aplicar la primera función sobre los elementos que ocupan posiciones impares y la segunda sobre los que ocupan posiciones pares.

Ejemplo:

```
> (aplicarFunciones (lambda(x) (1+ x)) (lambda(x) (1- x)) '(1 2 3 4 5))
```

```
(2 1 4 3 6)
```

5. Repite el ejercicio anterior, pero pasando los elementos en forma de número arbitrario de parámetros en lugar de una lista.

Ejemplo:

```
> (aplicarFunciones (lambda(x) (1+ x)) (lambda(x) (1- x)) 1 2 3 4 5)
```

```
(2 1 4 3 6)
```

6. Sin utilizar la función LIST, Implementa una función que emule su comportamiento. La función debe llamarse **listar**, admitir un número arbitrario de parámetros y devolver una lista con los parámetros

Ejemplos:

```
> (listar 2 3 4 5)
```

```
(2 3 4 5)
```

```
> (listar)
```

```
NIL
```