

# Ejercicios Paquete #1

Oscar Andrés Rosas Hernández. *Instituto Politécnico Nacional*

## I. 1

El quinto elemento de la lista `(( (1 2) 3) 4 (5 (6)) A (B C) D (E (F G)))`, sin usar la función `FIFTH`.

```
(print
  (first (rest (rest (rest (rest ' ( ((1 2) 3) 4 (5 (6)) A (B C) D (E (F G))))))))))

(print
  (fifth ' ( ((1 2) 3) 4 (5 (6)) A (B C) D (E (F G))))
)
```

El número de segundos que tiene el año bisiesto 2004.

```
(print
  ; days hour minute second
  (* 366 24 60 60)
)
```

Si el valor numérico asociado a la variable `x` es diferente de cero y además menor o igual que el valor asociado a la variable `y`

```
(setq x 10)
(setq y 9)

(print
  (and (not (eql 0 x)) (<= x y) )
)
```

Una lista con las dos soluciones reales de la ecuación

```
; 2x^2+7x+5=0

(setq a 3)
(setq b 5)
(setq c 2)

(print
  (list
    (/
      (+
        (- b)
        (sqrt
          (-
            (* b b)
            (* 4 a c) ) ) )
        (* 2 a))
    (/
      (-
        (- b)
        (sqrt
          (-
            (* b b)
            (* 4 a c) ) ) )
        (* 2 a)))
  )
```

## II. 2

Escriba, en notación prefija y evalúe las siguientes expresiones aritméticas:

■ a)

```
(print (+ (* 2 4) (- 6 8) ) )
```

■ b)

```
(print (/ (+ 5 -3 4) (+ 6 (/ 2 5) ) ) )
```

■ c)

```
(print
  (sqrt
    (/
      (+ (- (- 4 (/ 3 8) ) ) 1.4502)
      (expt -1 (expt (- 3 5) (/ 1 3) ) )
    )
  )
)
```

■ d)

```
(print
  (expt
    (/
      (expt
        (/ 65.402 (sqrt -1) )
        (/ 1 5)
      )
      0.17
    )
    (/ 1 7)
  )
)
```

## III. 3

- a) (TWO)
- b) ((EVA LISA) KARL SVEN EVA LISA KARL SVEN)
- c) (EVA GITAN LISA GITAN KARIN)
- d) (EVA LISA ANNA)
- e) (KARL ADAM NILSSON)
- f) C
- g) (C D E)
- h) (C B)
- i) (4)

## IV. 4

Defina una función Recombina que reciba como argumento una lista de la forma ((A . x) (B . y) (C . z)), donde A, B y C son átomos simbólicos, mientras que x, y y z son números. Como respuesta, la función debe entregar otra lista con la siguiente estructura: ( ((x y) . A) ((y z) . C) ((z y x) . B) )

```
(defun recombina
  (data)

  (let
    (
      (arg1 (first data))
      (arg2 (second data))
      (arg3 (third data))
    )
    (let
      (
        (A (first arg1))
        (B (first arg2))
        (C (first arg3))

        (x (rest arg1))
        (y (rest arg2))
        (z (rest arg3))
      )
    )
  )
```

```

    (let
      (
        (res1
          (cons (list x y) A))
        (res2
          (cons (list y z) C))
        (res3
          (cons (list z y x) B))
      )
      (list res1 res2 res3)
    )
  )
)

(print (recombina '((A . x) (B . y) (C . z)) ))

```

## V. 5

Defina un predicado RealNoCero? que reciba un argumento N y responda si su argumento es o no un número real diferente de cero.

```

(defun realNoCero(N)
  (and
    (not (eql N 0))
    (realp N)
  )
)

(print (realNoCero 'a))

```

## VI. 6

Construya una función Analiza, con argumento X, que responda una lista con los valores de verdad correspondientes a las respuestas a las siguientes preguntas: ¿es X un átomo?, ¿es X un número?, ¿es X una lista?, ¿es X una celda de construcción? y ¿es X una lista vacía?

```

(defun analiza(X)
  (list
    (atom X)
    (numberp X)
    (listp X)
    (consp X)
    (null X)
  )
)

(print (analiza NIL))

```

## VII. 7

Defina una función Intercala que reciba como argumentos dos listas cualesquiera y, como resultado entregue otra lista en la que se encuentran intercalados los elementos de las listas originales; siempre en el mismo orden: un elemento de la primera lista y otro de la segunda lista. Si las listas no tienen la misma longitud, todos los elementos restantes de la lista más grande se colocan seguidos en la respuesta.

```

(defun intercala
  (x y)
  (reverse
    (let
      (
        (result ())
        (total (+ (length x) (length y) ) )
      )
      (do
        (
          (i 0 (+ i 1))
          (j 0 (+ j 1))
        )
        ( (<= total (length result)) result )
        (if (eql (nth i x) nil) nil (push (nth i x) result) )
        (if (eql (nth j y) nil) nil (push (nth j y) result) )
      )
    )
  )
)

```

```

    )
  )
)
(print (intercala '(X Y) '(A B C D) ))

```

## VIII. 8

Programa un predicado `MismoTipo` que reciba como argumento dos listas de la misma longitud y como respuesta, devuelva `T` si ambas listas tienen elementos del mismo tipo y en las mismas posiciones, `NIL` en caso contrario. Observe que los elementos no requieren ser iguales, sólo del mismo tipo de datos.

```

(defun MiTipo
  (x y)
  (reverse
    (let
      (
        (result ())
        (total (length x))
      )
      (do
        ( (i 0 (+ i 1)) )
        ( (<= total (length result)) result )
        (push
          (eql (type-of (nth i x)) (type-of (nth i y)))
          result)
        )
      )
    )
  )
)
(print (MiTipo (list 'b 5 T) (list 'A 3 T) ))

```

## IX. 9

Defina una función `APalíndromo`, sensible a mayúsculas y minúsculas, que reciba como argumento una cadena y, como respuesta entregue otra cadena que es el palíndromo de la original. Ejemplo: `APalíndromo("Hola") = "HoloaloH"`.

```

(defun APalíndromo (str)
  (concatenate 'string str (reverse str))
)
(print (APalíndromo "Hola"))

```

## X. 10

Defina un predicado `Bisiesto` que reciba como entrada un número entero representando un año y, como respuesta, indique si se trata de un año bisiesto o no.

```

(defun Bisiesto (year)
  (or
    (and
      (zerop (mod year 4))
      (not (zerop (mod year 100)))
    )
    (zerop (mod year 400))
  )
)
(print (Bisiesto 2020))

```