

## $\lambda$ -calculus - Ejercicio 3.5

$$(\lambda x. ( (\lambda y. x y) z ) ) (\lambda x. x y)$$

$$(\lambda x. (\lambda y. x y) z) (\lambda x. x y)$$

Orden Normal	Orden Aplicativo
$(\lambda x. (\lambda y. x y) z) (\lambda x'. x' y')$ $(\lambda y. (\lambda x'. x' y') y) z$ $(\lambda x'. x' y') z$ $z y'$	$(\lambda x. (\lambda y. x y) z) (\lambda x'. x' y')$ $(\lambda x. x z) (\lambda x'. x' y')$ $(\lambda x'. x' y') z$ $z y'$

## $\lambda$ -calculus - Ejercicio 3.6

$$((\lambda y.(\lambda x.(\lambda x.\lambda y.x)x))y)MN$$

$$(\lambda y.(\lambda x.(\lambda xy.x)y)MN$$

Orden Normal	Orden Aplicative
$(\lambda y.(\lambda x.(\lambda x'y'.x')x)y)MN$	$(\lambda y.(\lambda x.(\lambda x'y'.x')y)MN$
$(\lambda x.(\lambda x'y'.x')x)MN$	$(\lambda y.(\lambda xy'.x)y)MN$
$(\lambda x'y'.x')MN$	$(\lambda yy'.y)MN$
$(\lambda y'.M)N$	$(\lambda y'.M)N$
$M$	$M$

## APL - Ejercicio 4

```
A ← (1 + ⌊ 3) , 3 + ⌊ 3
A [1 4]
2 4
A [A]
3 4 4 4 5 6
A [A , A]
3 4 4 4 5 6 3 4 4 4 5 6
A [⌊A ÷ 2]
2 2 3 3 3 4
```

1. Se asigna a A el vector **2 3 4 4 5 6** (porque se concatenan dos índices de 3 desplazados)
2. Se pide por los índices **1 4** que son **2 4**
3. Se pide por los índices **2 3 4 4 5 6** (A)
4. Se pide por los índices **2 3 4 4 5 6 2 3 4 4 5 6** (A duplicado)
5. Se pide por los índices **1 1 2 2 2 3** (todos los miembros, divididos por 2 y redondeados para abajo)

## APL - Ejercicio 5

```
B ← 'SIC TRANSIT' , 'GLORIA MUNDI'
ρ B
23
B [2 × ⍳ 3]
I R
B [1 + (ρ B) - ⍳ ρ B]
IDNUM AIROLGTISNART CIS
```

1. Se asigna a B la cadena 'SIC TRANSITGLORIA MUNDI'
2. Se pide el tamaño de la cadena (23 caracteres)
3. Se pide por los índices 2 4 6 (el índice de 3, multiplicado por 2), de los cuales se obtiene I, un espacio en blanco y R
4. Se pide por los índices del 23 al 1

## APL - Ejercicio 7

```
      4 5 ρ V ← 2 1 3 2 4 5 6 6 2 1
2 1 3 2 4
5 6 6 2 1
2 1 3 2 4
5 6 6 2 1
      T ← 3 3 4 ρ V
      , T
2 1 3 2 4 5 6 6 2 1 2 1 3 2 4 5 6 6 2 1 2 1 3 2 4 5 6 6 2 1 2 1 3 2 4 5
      ρ T
3 3 4
      ρ , T
36
```

1. Se asigna a V el vector **2 1 3 2 4 5 6 6 2 1**, y se imprime como una matriz de 4 filas y 5 columnas
2. Partiendo de V, se asigna a T un array de 3 dimensiones
3. Se imprime T planchado
4. Se pide la forma de T: 3 planos de 3 filas y 4 columnas
5. Se pide por las dimensiones del vector planchado:  $3 \times 3 \times 4 = 36$

## APL - Ejercicio 11g

El promedio entre el primer número positivo de un vector  $V$  y el último número negativo del mismo.

```
V ← -4 8 -15 16 -23 42
FIRST_POSITIVE ← V[(×V) ⍳ 1]
FIRST_POSITIVE
8
LAST_NEGATIVE ← (ϕV)[(ϕ(×V)) ⍳ -1]
LAST_NEGATIVE
-23
(FIRST_POSITIVE + LAST_NEGATIVE) ÷ 2
-7.5
```

## FP - Eliminar Vacíos

Dado una lista átomos con conjuntos vacíos intermedios, devolver solamente los átomos.

```
Def apndlst = apndr o [id, ~<>]
Def collect = null o 1 -> 2 ; apndl
Def ev = /collect o apndlst
ev: <1 2 3>
RESULT: <1, 2, 3>
ev: <<> 1 2 3 <>>
RESULT: <1, 2, 3>
ev: <<> 1 <> 2 <> 3 <>>
RESULT: <1, 2, 3>
ev: <1>
RESULT: <1>
ev: <>
RESULT: <>
```

## FP - 14A

Dada una secuencia de pares ordenados donde la primera componente indica el equipo que resultó ganador y la segunda indica el perdedor y donde cada par ordenado indica un partido jugado (no hay empates) obtener los equipos invictos.

```
Def in = /or o @eq o dist1
Def keep = in -> ~<> ; 1
Def ganadores = @1
Def perdedores = @2
Def diff = ev o @keep o distr
Def invictos = diff o [ganadores, perdedores]
invictos: <<1 2> <2 3> <4 2>>
RESULT: <1, 4>
```