



## Examen – Febrero de 2010 - 2ª parte

### Problema 1:

Ejercicio teórico.  
Ver material del curso.

### Problema 2:

a) 

```
function y = Pertenece( Conj, X )
n = length( Conj );
i = 1;
y = 0;
while y == 0 & i <= n
    if Conj( i ) == X
        y = 1;
    end
    i = i + 1;
end
```

b) 

```
function y = Agregar( Conj, X )
n = length( Conj );
i = 1;
salir = 0;
aux = Pertenece( Conj, X );
if n == 0
    y = X;
elseif X > Conj( n )
    y = [ Conj, X ];
else
    if aux == 0
        while salir == 0 & i <= n
            if X < Conj( i );
                y = [ Conj( 1:i - 1 ), X, Conj( i:n ) ];
                salir = 1;
            end
            i = i + 1;
        end
    else
        y = Conj;
    end
end
```



c) **function** y = Union( ConjA, ConjB )  
na = length( ConjA );  
nb = length( ConjB );  
**if** na == 0 | nb == 0  
    y = [ ConjA, ConjB ];  
**else**  
    **for** i = 1 : na  
        ConjB = Agregar( ConjB, ConjA( i ) );  
    **end**  
    y = ConjB;  
**end**

d) **function** y = Interseccion( ConjA, ConjB )  
na = length( ConjA );  
nb = length( ConjB );  
y = [ ];  
**if** na == 0 | nb == 0  
    y = [ ];  
**else**  
    **for** i = 1:na  
        aux = Pertenece( ConjB, ConjA( i ) );  
        **if** aux == 1  
            y = [ y, ConjA( i ) ];  
        **end**  
    **end**  
**end**

### Problema 3:

a) **function** [ ys,yi,yj,fil,col ] = espVertical( as,ai,aj,f,c )  
n = length( as );  
**if** n == 0  
    ys = [ ];  
    yi = [ ];  
    yj = [ ];  
    fil = f;  
    col = c;  
**else**  
    [aux1,aux2,aux3,aux4,aux5] =  
        espVertical( as(2:n), ai(2:n), aj(2:n), f, c );  
  
    ys = [ as( 1 ), aux1 ];  
    yi = [ ai( 1 ), aux2 ];  
    yj = [ c - ( aj( 1 ) - 1 ), aux3 ];  
    fil = aux4;  
    col = aux5;  
**end**



```
b) function [ ys, yi, yj, fil, col ] = rot90( as, ai, aj, f, c )
n = length( as );

if n == 0
    ys = [ ];
    yi = [ ];
    yj = [ ];
    fil = c;
    col = f;
else
    [ aux1,aux2,aux3,aux4,aux5 ] =
        rot90( as(2:n), ai(2:n), aj(2:n), f, c );

    ys = [ as( 1 ), aux1 ];
    yi = [ aj( 1 ), aux2 ];
    yj = [ f - ( ai( 1 ) - 1 ), aux3 ];
    fil = aux4;
    col = aux5;
end
```

#### Problema 4:

```
a) function M = tiros( N, S )
M = [ ];
for i = 1:N
    for j = 1:N
        for t = 1:N
            suma = i + j + t;
            if suma == S
                M = [ M; [ i, j, t ] ];
            end
        end
    end
end
```



```
b) function M = tirosGeneral( N, S, X )
    %X es la cantidad de veces que tiro el dado
    M = [ ];
    if X == 0 | S <= 0
        M = [ ];
    elseif X == 1
        if S <= N
            M = [ N ];
        end
    else
        for i = 1:N
            aux = tirosGeneral( N, S-i, X-1 );
            % Cuando existe por lo menos una forma de sumar S-i
            % tirando X-1 veces
            % agrego una columna donde todos sus elementos
            % valen i a la izquierda de aux.
            % Por último agrego el nuevo valor de aux debajo de
            % la matriz M.
        end
    end
end
```