

Pregunta 1:

b. Una constante puede aparecer en la parte izquierda y derecha de una asignación.

Pregunta 2:

b. La matriz A se representa en forma dispersa en forma elemental como $f=[1,2,3,3,4,4,5]$, $c=[1,2,2,3,3,4,5]$ y $d=[1,2,-5,3,-2,1,4]$.

Pregunta 3:

a. 18_{10} .

Pregunta 4:

b. Todos los sistemas operativos son monousuario.

Pregunta 5:

b. Da error porque está mal usada la sentencia de asignación.

Pregunta 6:

a. La memoria ROM es únicamente de lectura.

Pregunta 7:

a. 1 y 2.

Pregunta 8:

b. 01101

Pregunta 9:

b. Un número desnormalizado.

Pregunta 10:

b. 0 1000010 000000000000000000000000

Solución Ejercicio 1:

```
function K = kronecker(A, B)
[m,n] = size(A);
[p,q] = size(B);
K = [];
for i=1:m
    nueva = [];
    for j=1:n
        nueva = [nueva, A(i,j)*B];
    end
    K = [K; nueva];
end
```

Solución Ejercicio 1 (Alternativa):

```
function K = kronecker(A, B)
[m,n] = size(A);
[p,q] = size(B);
K = zeros(m*p, n*q);
for i=1:m
    for j=1:n
        for x=1:p
            for y=1:q
                fila = (i-1)*p + x;
                col = (j-1)*q + y;
                K(fila, col) = A(i,j)*B(x,y);
            end
        end
    end
end
```

Solución Ejercicio 2:

Parte a)

```
function cant = contar(v, min, max)
len = length(v);
if (len == 0)
    cant = 0;
else
    if (min <= v(1) && v(1) <= max)
        cant = 1 + contar(v(2:len), min, max);
    else
        cant = contar(v(2:len), min, max);
    end
end
```

Parte b)

```
function vcant = vcontar(M, min, max)
    [filas, columnas] = size(M);
    if columnas == 0
        vcant = [];
    else
        vcant = [contar(M(:,1), min, max), vcontar(M(:,2:columnas), min, max)];
    end
```

Parte c)

```
function [colMin, colMax] = maxMinColumns(M, min, max)
    auxVec = vcontar(M, min, max);
    len = length(auxVec);
    if len == 0
        colMin = 0;
        colMax = 0;
    else
        colMin = 1;
        colMax = 1;
        for i = 2:len
            if(auxVec(i) < auxVec(colMin))
                colMin = i;
            elseif(auxVec(i) > auxVec(colMax))
                colMax = i;
            end
        end
    end
end
```

Solución Ejercicio 3:

Parte a)

```
function v = numToBin(n)
    if n < 2
        v = [n];
    else
        resto = mod(n, 2);
        n = floor(n/2);
        v = [numToBin(n), resto];
    end
```

Parte b)

```
function n = binToNum(v)
    len = length(v);
    n = 0;
    for i=1:len
        n = n + v(i) * 2^(len-i);
    end
```

Parte c)

```
function r = largoN(v, n)
    len = length(v);
    if len < n
        r = [zeros(1, n-len), v];
    else
        r = v((len-n+1): len);
    end
```

Parte d)

```
function res = opRara(a, b)
    binA = numToBin(a);
    binB = numToBin(b);
    lenBinA = length(binA);
    lenBinB = length(binB);

    if (lenBinA < lenBinB)
        binA = largoN(binA, lenBinB);
        len = lenBinB;
    else
        binB = largoN(binB, lenBinA);
        len = lenBinA;
    end

    binRes = [];
    for i=1:len
        binRes(i) = xor (binA(i), binB(i));
    end

    res = binToNum(binRes);
```