

1) Calcule y explique el resultado de pasar el número -101 (representado en base 10) a:

a) binario con signo 8 bits :

$101_{10} = 1\ 1100101$  sí se puede representar.

b) complemento a 1 con 7 bits:

$101_{10} = 1100101_{C1}$ , ya tiene 7 bits y un 1 en el bit más significativo, significa que representa un negativo, pero 101 es positivo. Entonces, no se puede representar

Tampoco se puede representar  $-101_{10} = 0011010_{C1}$  que representa un positivo.

c) complemento a 2 con 7 bits.

Tampoco se puede representar porque  $-101_{10} = 0011011_{C2}$  que representa un positivo.

d) Corrimiento a M con 7 bits (con  $M = 2^n - 1$ ).

$M=64$ , entonces  $-101+64 = -37 < 0$ , por lo que no se puede representar en exceso a M con 7 bits.

e) Punto flotante con 1 bit de signo, 4 bits de exponente y 6 bits de mantisa

$-101_{10} = -1100101_2 = -1,100101_2 * 2^6 = 1\ 1101\ 100101$  sí se puede representar.

$M = 2^{4-1} - 1 = 8 - 1 = 7 \Rightarrow 7 + 6 = 13 = 1101$

2) Si a y b están en binario, octal o base 10, el código es exactamente el mismo.

```
function res = esmayor(a,b)
    la=length(a);
    lb=length(b);
    if la>lb
        b=[zeros(1,la-lb) b];
        lb=la;
    else
        a=[zeros(1,lb-la) a];
        la=lb;
    endif
    i=1;
    while i<=la && a(i)==b(i)
        i=i+1;
    endwhile
    res=0;
    if i<=la && a(i)>b(i)
        res=1;
    endif
endfunction
```

3a)

```
function res=DivisoresR(n,v)
    lv=length(v);
    if lv==0
        res=[];
    else
        res=DivisoresR(n,v(2:lv));
        if mod(n,v(1))==0
            res=[v(1) res];
        end
    end
endfunction
```

3b)

```
function pri=halloPrimosR(n)
    if n==2
        pri=2;
    else
        pri=halloPrimosR(n-1);
        d=DivisoresR(n,pri);
        if length(d)==0 %n sólo se divide por 1 y por sí mismo.
            pri=[pri n];
        end
    end
endfunction
```

4a) Se resuelve con while, porque los vectores de M están ordenados por filas

```
function sf=SumFils(Mv,Mf,Mc,m)
    lM=length(Mv);
    sf=0;
    i=1;
    while i<=lM && Mf(i)<=m
        sf=sf + Mv(i);
        i=i+1;
    endwhile
endfunction
```

4b) Se resuelve con for, porque como está ordenados por filas y dentro de ellas por columnas, en sí los números de columna no están ordenados.

```
function sc=SumCols(Mv,Mf,Mc,n)
    lM=length(Mv);
    sc=0;
    for i=1:lM
        if Mc(i)<=n
            sc=sc + Mv(i);
        endif
    endfor
endfunction
```

5)

```
function res=RestoUnoR(b)
    lb=length(b);
    if b(lb)==1
        res=b;
        res(lb)=0;
    else
        res=[RestoUnoR(b(1:lb-1)) 1];
    endif
endfunction
```

6a)

```
function res=prefR(v,p)
    lv=length(v); lp=length(p);
    if lv<lp
        res=0;
    elseif lp==0
        res=1;
    elseif v(1)~=p(1)
        res=0;
    else
        res=prefR(v(2:lv),p(2:lp));
    endif
endfunction
```

6b)

```
function res = contR(v,c)
    lv = length(v); lc = length(c);
    if lc==0
        res = 1;
    elseif lc > lv
        res = 0;
    elseif prefR(v,c)
        res = 1;
    else
        res = contR(v(2:lv),c);
    endif
endfunction
```

6c)

```
function res = contMit(M,c)
    [fM,cM]=size(M); lc=length(c);
    if lc==0
        res=1;
    elseif lc>fM && lc>cM
        res=0;
    else
        i=1; res=0;
        while i<=fM && ~res
            res = contR(M(i,:),c);
            i=i+1;
        endwhile
        i=1;
        while i<=cM && ~res
            res = contR(M(:,i)',c);% Puede no ponerse la transpuesta que igual
            i=i+1; % va a funcionar.
        endwhile
    end
endfunction
```