

Examen – 12 de febrero de 2025

|            |                    |  |
|------------|--------------------|--|
| Problema 1 | 12 (2,2,2,3,3) pts |  |
|------------|--------------------|--|

Justifique sus respuestas:

a) Agrupamos de 4 en 4 desde la derecha:

1010 1100 1101 1100

Cada grupo se convierte a hexadecimal:

- $1010_2 = A_{16}$
- $1100_2 = C_{16}$
- $1101_2 = D_{16}$
- $1100_2 = C_{16}$

Resultado:

ACDC<sub>16</sub>

b)

-0110111 → Ca1: 1001000

-0001001 → Ca1: 1110110

1001000 + 1110110 = 0111111 : el resultado es un número positivo así que hay desbordamiento

c)

-0110111 → Ca1: 1001000 → Ca2: 1001001

-0001001 → Ca1: 1110110 → Ca2: 1110110

1001001 + 1110111 = 1000000

d)  $256 = 2^8$  y  $1 = 2^0$ . La diferencia entre ambos exponentes es 8, superior a la cantidad de bits de la mantisa, por lo que se pierden todos los bits de mantisa del "1" (incluyendo el 1 implícito) y el resultado de la suma es 256 las dos veces.

e) Para igualar los exponentes se corre la mantisa a la derecha 7 lugares. Por lo que la suma de mantisas queda:  $1,0000000 + 0,0000001 = 1,0000001$ , es decir, no se pierden bits y el resultado es 258.

|            |       |  |
|------------|-------|--|
| Problema 2 | 6 pts |  |
|------------|-------|--|

aux\_func(1)=1+1=2

aux\_func(2)=2+1=3

aux\_func(3)=3-aux\_func(2)=3-3=0

Ejecución de miscript.m

1. Inicialización:

- $x=3$
- $y=1$

2. Primer ciclo (z=1):

- Calculamos  $y=aux\_func(y)+z=aux\_func(1)+1=2+1=3$ .

- Actualizamos  $x=x+1=4$ .
3. Segundo ciclo ( $z=2$ ):
- Calculamos  $y=\text{aux\_func}(y)+z=\text{aux\_func}(3)+2=0+2=2$ .
  - Actualizamos  $x=x+1=5$ .

|            |                |  |
|------------|----------------|--|
| Problema 3 | 22 (12,10)ptos |  |
|------------|----------------|--|

Parte a)

```
function suma = sumaDigitosPares(n)
    suma = 0;
    while n > 0
        digito = mod(n, 10); % Obtener el último dígito
        if mod(digito, 2) == 0 % Verificar si es par
            suma += digito;
        end
        n = floor(n / 10); % Quitar el último dígito
    endwhile
end
```

Parte b)

```
function c = logEntRec(a, b)
    if a < b
        c = 0; % Si a es menor que b, el logaritmo entero es 0
    else
        c = 1 + logRec(floor(a / b), b);
    end
end
```

|            |                 |  |
|------------|-----------------|--|
| Problema 4 | 24 (12,12) ptos |  |
|------------|-----------------|--|

a)

```
function [Rf, Rc, Rd] = menoresQueN(Mf, Mc, Md, n)
```

```
Rf=[]; Rc=[]; Rd=[];
```

```
for i = 1:length(Mf)
    if Md(i) < n
        Rf=[Rf,Mf(i)];
        Rc=[Rc,Mc(i)];
        Rd=[Rd,Md(i)];
    end
end
```

end

b)

```
function [Rf, Mc, Md] = intercambiarFilas(Mf, Mc, Md, i, j)
```

```
n = length(Mf);
if n==0
    Rf = [];
else
    [Rf, Mc, Md] = intercambiarFilas(Mf(2:n), Mc, Md, i, j);

    if Mf(1) == i
        Rf = [j,Rf];
    elseif Mf(1) == j
        Rf = [i,Rf];
    end
end
```

```

else
    Rf = [Mf(1),Rf];
endif
endif
end

```

|            |                       |  |
|------------|-----------------------|--|
| Problema 5 | 24<br>(12,12)Pt<br>os |  |
|------------|-----------------------|--|

a)  
function s = buscarRango(v, a, b)

```

s=[];
i=1;

while i <= length(v) && v(i)<a
    i=i+1;
end

while i <= length(v) && v(i)<=b
    s = [s, v(i)];
    i=i+1;
end
end

```

b)  
function [sec,signo] = delimitarSecuencias(v)

```

n = length(v);
signoActual=9999; %cualquier valor no válido
sec=[];
signo=[];

if n >= 2
    for i = 1:n-1
        if v(i) < v(i+1)
            nuevoSigno=1;
        elseif v(i) == v(i+1)
            nuevoSigno=0;
        else
            nuevoSigno=-1;
        end

        if signoActual ~= nuevoSigno
            signoActual=nuevoSigno;
            sec=[sec,i];
            signo=[signo,nuevoSigno];
        end
    end
end
end
end

```

|            |         |  |
|------------|---------|--|
| Problema 6 | 12 Ptos |  |
|------------|---------|--|

```

function w = intersecRec(u, v)
    lu=length(u);

```

```
lv=length(v);  
if u==0 || v==0  
    w = [];  
else  
    if u(1) == v(1)  
        w = intersecRec(u(2:lv), v(2:lv));  
        if length(w)==0 || u(1)~=w(1)  
            w=[u(1),w];  
        endif  
    elseif u(1) < v(1)  
        w = intersecRec(u(2:lv), v);  
    else  
        w = intersecRec(u, v(2:lv));  
    end  
end  
end  
end
```