

Examen – 22 de julio de 2024

Problema 1	12 (3,3,3,3) ptos	
-------------------	-------------------	--

- a) $2^{13} - 2^0$ se representa en binario como una tira de 13 unos. A este número se le debe restar 2^6 para generar un 0 en la 7ma posición. Por lo tanto las respuestas correctas son a=13, b y c=6 y 0 o 0 y 6.
- b) La suma se puede realizar directamente usando el algoritmo habitual. $B+2=D$, $C+1=D$, $0+9=9$, y $A+1=B$. Ninguno de los dígitos genera acarreo, por lo tanto el resultado es $B9DD_{16}$.
- c) La tira binaria representa el número 5, pero al estar en exceso $M=128$, es necesario restar 128 al número representado. Luego se debe representar el resultado (-123) en complemento a 2.
 $123=64+32+16+8+2+1=1111011_2$. Para obtener la representación de (-123) en complemento a 2 de 8 bits se agrega un 0 a la izquierda, se complementa el número y se suma 1, obteniendo 10000101.
- d) En binario $2^2+2^{-1}+2^{-3}$ se representa como 100,101 que equivale a $1,00101 \times 2^2$. Por lo tanto el exponente se representa como $2+7=9$, es decir 1001, y la mantisa es 0010100 (completando 7 bits), por lo que el número queda 010010010100.

Problema 2	12 (8,4) ptos	
-------------------	---------------	--

A)

```
function res = buscarDigito(n, d)
    res = 0;
    while n > 0 && res==0
        if mod(n, 10) == d
            res = 1;
        else
            n = floor(n / 10);
        end
    end
end
```

End

B)

```
function res = filtrarCedulas(v, d)
    res = [];
    for i = 1:length(v)
        if buscarDigito(v(i),d)
            res = [res, v(i)];
        end
    end
end
```

```

end
end
end

```

Problema 3	6 ptos	
-------------------	--------	--

- La variable b recibe el valor de salida de la función recursiva *funcAux* que al alcanzar el paso base devuelve el mismo valor i que se le pasa inicialmente por parámetro. Por lo tanto $b=5$.
- La variable v es un vector cuyo 3er elemento aumenta en 1, es decir, $v=[12,7,-2,1]$.
- La variable i comienza valiendo 5 y se decrementa en la 4ta línea, es decir, $i=4$.

Problema 4	34 (8,12,14) ptos	
-------------------	-------------------	--

A)

```

function ind = encontrarIndice(v, x)
    i = 1;
    while i <= length(v) && v(i) < x
        i = i + 1;
    end
    if i > length(v)
        ind = -1;
    else
        ind = i;
    end
end
end

```

B)

```

function ord_v = ordenarVector(v)
    % Inicializar el vector ordenado vacío
    ord_v = [];

    % Iterar sobre cada elemento del vector original
    for i = 1:length(v)
        lord = length(ord_v);
        % Encontrar el lugar adecuado para insertar el elemento actual en ord_v
        pos = encontrarIndice(ord_v, v(i));

        if pos == -1

```



```
        % todos son menores, insertar al final de ord_v
        ord_v = [ord_v, v(i)];
    else
        % insertar antes que el primer elem mayor o igual
        ord_v = [ord_v(1:pos-1), v(i), ord_v(pos:lord)];
    end
end
end
```

end

C)

% Opción con un solo while:

```
function [ini, fin] = encontrarSecuencias(v)
```

```
    ini = [];
```

```
    fin = [];
```

```
    n = length(v);
```

```
    i = 1;
```

```
    f = 1;
```

```
    while i < n
```

```
        if f < n && v(f) + 1 == v(f + 1)
```

```
            f = f + 1;
```

```
        else
```

```
            if f > i
```

```
                ini = [ini, i];
```

```
                fin = [fin, f];
```

```
            end
```

```
            i = f + 1;
```

```
            f = i;
```

```
        end
```

```
    end
```

end

% Opción con dos while anidados:

```
function [ini, fin] = encontrarSecuencias(v)
```

```
    ini = [];
```

```
    fin = [];
```

```
    n = length(v);
```

```
i = 1;

while i < n
    % buscar el final de la secuencia ascendente que comienza en i
    f = i;
    while f < n && v(f) + 1 == v(f + 1)
        f = f + 1;
    end
    if f > i
        ini = [ini, i];
        fin = [fin, f];
    end
    % Continuar la búsqueda desde el próximo elemento
    i = f + 1;
end
end
```

Problema 5	22 (12,10) ptos	
-------------------	-----------------	--

A)

```
function [fil2, col2, val2] = extraerCuadrante(c, fil, col, val)
    n=length(fil);
    if n==0
        fil2=[]; col2=[]; val2=[];
    else
        [fil2, col2, val2] = extraerCuadrante(c, fil(2:n), col(2:n), val(2:n));
        if fil(1)<=c && col(1)<=c
            fil2=[fil(1),fil2];
            col2=[col(1),col2];
            val2=[val(1),val2];
        end
    end
end
```

B)

```
function nnzf = nnzPorFila(filas, columnas, valores, f)
    nnzf = zeros(1,f);

    for i = 1:length(filas)
        nnzf(filas(i))=nnzf(filas(i))+1;
    end
end
```

Problema 6	14 ptos	
-------------------	---------	--

```
function camino = laberinto(L)
    [m,n]=size(L);
    if (L(1,1)==1 || L(m,n)==1)
        camino = [];
    elseif (m==1 && n==2)
        camino = [0];
    elseif (m==2 && n==1)
        camino = [1];
    else
        camino = laberinto(L(:,2:n)); % muevo a la izquierda
        if length(camino)~=0
            camino = [0,camino];
        else
            camino = laberinto(L(2:m,:)); % muevo hacia abajo
            if length(camino)~=0
                camino = [1,camino];
            end
        end
    end
end
```