

2^{do} Parcial - 28 de noviembre de 2019

Problema 1	8 (1, 1, 2, 2, 2) ptos	
-------------------	------------------------	--

a) Número 15 en punto fijo con 5 bits: 01111

Complemento bit a bit de 15: 10000

-15 en complemento a 1 de 5 bits: 10000

b) $10011 + 10010 = 00101$. La suma no es representable en complemento a 2 con 5 bits.

c)

I. 4 y 8 son dos potencias de 2 consecutivas. De hecho 4 es 2^2 por lo que su representación en el sistema de PF planteado es 0 101 00000 y 8 es 2^3 por lo que su representación en el sistema de PF planteado es 0 110 00000 (con 3 bits de exponente el exceso $M=2^{3-1}-1=3$). La cantidad de números representables entre 4 y 8 es entonces 2 elevado a la cantidad de bits de la mantisa (5) + 1 (porque se incluyen los extremos).

II. El bit de signo es 1, por lo que se trata de un número negativo.

El exponente es 101, que equivale a 5. Restando el exceso (3) tenemos que el exponente es 2.

Por último, la mantisa es 10000, por lo que el número es $-1,1_2 \times 2^2 = -110_2 \times 2^0 = -6$.

III. La diferencia entre exponentes es 2, por lo que se debe desplazar dos lugares la mantisa del número de menor exponente introduciendo el "1," implícito. El resultado quedará con el exponente más grande (110). Como el sustraendo es negativo, lo negamos y simplemente sumamos las mantisas:

$$1,11101 + 0,01100 = 10,01001$$

Para normalizar, desplazamos la coma hacia la izquierda aumentando en 1 el exponente y perdiendo el último bit de la mantisa:

$$0\ 111\ 00100$$

El resultado no es representable en el sistema de PF planteado, ya que el exponente 111 se reserva para Inf y NaN

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

a)

```
function resultado = Suma(v)
    lv = length(v);
    if lv == 0
        resultado = 0;
    else
        resultado = v(1) + Suma(v(2:lv));
    endif
endfunction
```

b)

```
function [pares,impares] = SumaParImpar(v)
    lv = length(v);
    if lv == 0
        pares = 0;
        impares = 0;
    elseif lv == 1
        pares = 0;
        impares = v(1);
    else
        [pares,impares] = SumaParImpar(v(3:lv));
        impares = impares + v(1);
        pares = pares + v(2);
    endif
endfunction
```

Problema 3	8 ptos	
-------------------	--------	--

```
function resultado = esSimetrica(M)
    i = 1;
    resultado = true;
    while resultado && i <= size(M,1)
        j = i+1;
        while resultado && j <= size(M,2)
            if M(i,j) == M(j,i)
                j = j+1;
            else
                resultado = false;
            endif
        endwhile
        i = i+1;
    endwhile
endfunction
```

Problema 4	8 ptos	
-------------------	--------	--

```
function u = sumaReflexiones_rec(v)
    lv = length(v)
    if lv == 0
        u = [];
    else
        u = [v(1)+v(lv), sumaReflexiones_rec(v(2:lv-1))];
    endif
endfunction
```

Problema 5	8 ptos	
-------------------	--------	--

```
function M = Combinaciones(k)

    if (k == 0)
        M = [];
    else
        M = [0 1];
    endif

    for i = 2:k
        M = [zeros(1,size(M,2)) ones(1,size(M,2)); M M];
    endfor

endfunction
```

Problema 6	16 (8,8) ptos	
-------------------	---------------	--

a)

```
function y = spmv_it(f,c,d,x,m)
    y = zeros(m,1);
    for i=1:length(d)
        y(f(i)) = y(f(i)) + d(i)*x(c(i))
    endfor
endfunction
```

b)

```
function [f2,c2,d2] = mult_col_rec(f1,c1,d1,v)

    n = length(d1);

    if n == 0
        f2=[];
        c2=[];
        d2=[];
    else
        [f2,c2,d2]=mult_col_rec(f1(2:n),c1(2:n),d1(2:n),v);
        if v(c1(1)) ~= 0
            d2 = [d1(1)*v(c1(1)),d2];
            f2 = [f1(1),f2];
            c2 = [c1(1),c2];
        endif
    endif

endfunction
```