

Prueba Final - 10 de diciembre de 2020

Problema 1	8 (1, 1, 1, 1, 1, 3) ptos	
-------------------	---------------------------	--

- a) Hay dos representaciones del cero. Una es 00000000 y otra es 10000000.
- b) El menor número que se puede representar es -127 y se representa como 10000000.
- c) El mayor número que se puede representar es 127 y se representa como 01111111.
- d) El menor número que se puede representar es -128 y se representa como 10000000.
- e) El mayor número que se puede representar es 127 y se representa como 01111111.
- f) La tira 010001001001 representa el número $1,1001001x2^{exp}$, donde $exp=1000_2-M$ y M es el exceso de la representación. $4=2^2$ por lo que el resultado es $1,1001001x(2^{exp}/2^2)=1,1001001x2^{exp-2}$.
- $exp-2=1000_2-0010_2-M=8-2-M=6-M=0110_2-M$ para cualquier exceso M que permita representar ambos exponentes con 4 bits (por ejemplo $M=7$). Por lo tanto la representación de $\pi/4$ es 001101001001

Problema 2	13 (4,9) ptos	
-------------------	---------------	--

- a) El valor de la variable `a` luego de la invocación es 52.
- b) El valor de la variable `d` luego de la invocación es 10. Las variables `b` y `c` tienen el mismo valor que tenían previo a la invocación de `otraF`, si es que estaban definidas. Si no estaban definidas, quedan indefinidas. Esto se debe a que las variables `b` y `c` de la función `otraF` son locales a la misma.

Problema 3	10 ptos	
-------------------	---------	--

```
function res = sumaDigitos(x)
    res = 0;
    while (x ~= 0)
        res = res + mod(x,10);
        x = floor(x/10);
    endwhile
end
```

Problema 4	17 (8,9) ptos	
-------------------	---------------	--

a)

```
function primer = primerCero(x,fx)
    iter = 1;
    largo = length(fx);
    while (iter <= largo && fx(iter) ~= 0)
        iter = iter + 1;
    end
    if (iter > largo)
        primer = -1;
    else
        primer = x(iter);
    end
end
```

b)

```
function v_primer = primerCeroMatriz(x, Mx)
    n = size(Mx, 1);
    v_primer = zeros(1,n);
    for i = 1:n
        v_primer(i) = primerCero(x, Mx(i,:));
    end
end
```

Problema 5	12 pts	
-------------------	--------	--

```
function [min, max] = minMax(x)
    lx = length(x)
    if (lx == 1)
        min = x(1);
        max = x(1);
    else
        [min, max] = minMax(x(2:lx));
        if (min > x(1))
            min = x(1);
        elseif (max < x(1))
            max = x(1);
        end
    end
end
end
```

Problema 6	12 pts	
-------------------	--------	--

```
function v = min2Vect(x,y)
    lx=length(x);
    ly=length(y);
    if (lx == 0)
        v = [];
    else
        v_rec = min2Vect(x(2:lx),y(2:ly));
        if (x(1) < y(1))
            v = [x(1) v_rec];
        else
            v = [y(1) v_rec];
        end
    end
end
end
```

Problema 7	14 pts	
-------------------	--------	--

```
function b = alMenos(M, valor, cantidad)

    acumulador = 0;
    b = 0;

    i = 1;
    while (i <= size(M,1) && acumulador < cantidad)
        j = 1;
        while (j <= size(M,2) && acumulador < cantidad)
            if (M(i,j) == valor)
                acumulador = acumulador + 1;
            end
            j = j + 1;
        end
        i = i + 1;
    end

    if (acumulador == cantidad)
        b = 1;
    end
end
```

Problema 8	14 pts
-------------------	--------

```
function [fs, cs, vs] = sumaMatDispRec(fa, ca, va, fb, cb, vb)

    la=length(fa);
    lb=length(fb);

    if (la == 0 || lb == 0)
        fs = [fa, fb];
        cs = [ca, cb];
        vs = [va, vb];
    else
        if ((fa(1) < fb(1)) || (fa(1) == fb(1) && ca(1) < cb(1)))
            [fs, cs, vs] = sumaMatDispRec(fa(2:la), ca(2:la), va(2:la), fb, cb, vb);
            fs = [fa(1), fs];
            cs = [ca(1), cs];
            vs = [va(1), vs];
        elseif ((fb(1) < fa(1)) || (fb(1) == fa(1) && cb(1) < ca(1)))
            [fs, cs, vs] = sumaMatDispRec(fa, ca, va, fb(2:lb), cb(2:lb), vb(2:lb));
            fs = [fb(1), fs];
            cs = [cb(1), cs];
            vs = [vb(1), vs];
        elseif (fa(1) == fb(1) && ca(1) == cb(1))
            [fs, cs, vs] = sumaMatDispRec(fa(2:la), ca(2:la), va(2:la),
                                           fb(2:lb), cb(2:lb), vb(2:lb));

            if (va(1) + vb(1) ~= 0)
                fs = [fa(1), fs];
                cs = [ca(1), cs];
                vs = [va(1) + vb(1), vs];
            end
        end
    end
end
end
end
```