Examen - 19 de febrero de 2020
Solución

Problema 1	8 (1, 1, 1, 1, 4) ptos	

a) $F5A_{16} = 111\ 101\ 011\ 010_2 = 7532_8$.

b) $2343_7 * 10_7 = 23430_7$

c) En Ca2 de 4 bits 1111 representa al -1, por lo tanto la operación es correcta.

d)

-x	Ca2 de 4 bits
0	0000
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000
-9	No es representable en Ca2 de 4 bits

e) En el sistema planteado `a=1.0x2^7` se representa como 0 1110 0000000 y `b=1.0x2^-2` se representa como 0 0101 0000000. Para sumar a+b se debe llevar b al exponente de a:

0 1110 1,0000000

+ 0 1110 0,000000001

0 1110 1,000000001

Al representar el resultado se mantienen únicamente los 8 bits más significativos de la mantisa (normalizando luego), por lo que el resultado es 0 1110 0000000 y `a` no cambia su valor durante las 1000 iteraciones.

Problema 2	7 (1.5,5.5) ptos	

a) `minM` y `maxM` no cambian su valor durante la iteración, por lo que la función devuelve `x=1, y=1.`

b)

```
1  - function [minM,maxM] = minMaxMatriz(M)
2  -   [nf,nc]=size(M);
3  -   minM=M(1,1);
4  -   maxM=minM;
5  -   for i=1:nf
6  -     [min,max]=minMax(M(i,:));
7  -     if (minM > min)
8  -        minM = min;
9  -     end
10 -     if (maxM < max)
11 -        maxM = max;
12 -     end
13 -   end
14 - end
```

Problema 3	7 ptos	

Determine el valor de las variables `a`,`b` y `c` luego de ejecutar `miscript.m` desde la línea de comandos de octave.

```
% funcAux.m

function c = funcAux(a,b)
  if a<5
    c = 1;
  else
    c = b + funcAux(a-1,b);
  end
end
```

```
% miscript.m

c = 10;
b = funcAux(8,2);
a = b + 1;
```

c = 10
b = funcAux(8,2) = 2+funcAux(7,2) = 2+2+funcAux(6,2) = 2+2+2+funcAux(5,2) = 2+2+2+2+funcAux(4,2) = 2+2+2+2+1 = 9
a = 9+1 = 10

Problema 4	14 ptos	

```
function res = comienzaCon(v1, v2)
  n = length(v1);
  m = length(v2);
  if m > n
    res = 0;
  else % m <= n
    i = 1;
    res = 1;
    while i <= m && res
      res = v1(i) == v2(i);
      i = i + 1;
    end
  end
end
```

Problema 5	14 ptos	

```
function resultado = contarOcurrencias(valores, n)
   resultado = zeros(1,n);
   m = length(valores);
   for i = 1:m
      resultado(valores(i)) = resultado(valores(i)) + 1;
   end
end
```

Problema 6	14 ptos	

```
function res = pares(M)
   res = [];
   [m,n] = size(M);
   for i = 1:m    % se pueden invertir los fors
      for j = 1:n
           if mod(M(i,j),2) == 0
              res = [res, M(i,j)];
           end
      end
   end
end
```

Problema 7	12 ptos	

```
function x = binomial(n,k)
    if (n < k) || (n < 0) || (k < 0)
        x = 0;
    elseif (k == 0) || (n == k)
        x = 1;
    else
        x = binomial(n-1,k-1) + binomial(n-1,k);
    end
end
```

Problema 8	12 ptos	

```
function res = comienzaCon(v1, v2)
 n = length(v1);
 m = length(v2);
 if m > n
   res = 0;
 elseif m == 0
   res = 1;
 elseif v1(1) == v2(1)
   res = comienzaCon(v1(2:n), v2(2:m));
 else
    res = 0;
 end
end
```

Problema 9	12 ptos	

```
function [fd, cd, dd] = multiplicacionElementoaElementoFiltrada(A, B,
                                                                fs, cs, ds)
   l = length(fs);
   if l == 0
      fd = [];
      cd = [];
      dd = [];
   else
      [fd, cd, dd] = multiplicacionElementoaElementoFiltrada(A, B,
                        fs(2:l), cs(2:l), ds(2:l));
      prod = A(fs(1),cs(1)) * B(fs(1),cs(1));
      if (prod ~= 0)
         fd = [fs(1) fd];
         cd = [cd(1) cd];
         dd = [prod dd];
```

```
    endif
endif
```