Examen febrero 2019 - Computación 1

Problema 1

a) $11110_2 \rightarrow 30_{10}$

b) $101111000101_2 \rightarrow 5705_8$

c) $16_{10} \rightarrow$ el rango de representación de complemento a 1 con 5 bits es: [-15 , 15].
      No es posible representar 16 en complemento a 1 con 5 bits.

d) $67_{10} \rightarrow 1000011_2 = 1000011_2 \times 2^0 = 1{,}000011_2 \times 2^6$
      Exponente: $M = 2^7 - 1 = 127$
      $E = 6 + 127 = 133_{10} = 10000101_2$

      La tira de 32 bits queda:
      1 10000101 00001100000000000000000

e)   1001 + 0111 = 0001

Problema 2

a)
```
function y = minmaxVectRec( v, x )
     n = length(v);

     if v(1) > x
          y = x;
     elseif n==1
          y = v(1);
     else
          y = minmaxVectRec( v(2:n) , x );

          if v(1) > y
               y = v(1);
          end
     end
end
```

b)
```
function y = minmaxMatRec( M, x )
     [m,n] = size(M);
     if m == 1
          y = minmaxVectRec(M, x);
     else
          y = minmaxVectRec(M(1,1:n), x);

          if x > y
               yResto = minmaxMatRec(M(2:m,1:n), x);
               if yResto > y
                    y = yResto;
               end
          end
     end
end
```

c)
```
function y = minmaxMatIt( M, x )
     [m,n] = size(M);

     y = M(1,1);

     i = 1;
     while i <= m && y ~= x
          j = 1;
          while j <= n && y ~= x

               if M(i,j) > x
                    y = x;
               else
                    if M(i,j) > y
                         y = M(i,j);
                    end
               end

               j = j+1;
          end
          i = i+1;
     end

end
```

Problema 3

a)
```
function y = polEvalRec(P,x)

    lP = length(P);

    if lP == 0
        y = 0;
    elseif lP == 1
        y = P;
    else
        y = x * polEvalRec(P(1:lP-1),x) + P(lP);
    end
end
```

b)
```
function r = aproxRaiz(P,a,b,t)

    if (b-a)/2 <= t

        r = (a+b)/2;
    else
        m = (a+b)/2;
        Pm = polEvalRec(P,m);

        if Pm == 0
            r = m
        elseif polEvalRec(P,a)/Pm < 0
            r = aproxRaiz(P,a,m,t);
        else
            r = aproxRaiz(P,m,b,t);
        end

    end
end
```

Problema 4

a)
```
function d = detIt(M)

    n = size(M,1);
    if n==0
        d = 1;
    else
        fkm2 = 1;
        fkm1 = M(1,1);

        for k = 2 : n
            aux = fkm1;
            fkm1 = M(k,k) * fkm1 - M(k-1,k) * M(k,k-1) * fkm2;
            fkm2 = aux;
        end
        d = fkm1;
    end
end
```

b)
```
function d = detRec(M)
    n = size(M,1);
    if n == 0
        d = 1;
    elseif n == 1
        d = M;
    else
        d= M(n,n)*detRec(M(1:n-1,1:n-1))-
                M(n-1,n)* M(n,n-1)*detRec(M(1:n-2,1:n-2));
    end
end
```

Problema 5
a)
```
function y = esTri(Mn,Mi,Mj)

    % Mn valores
    % Mi índices de filas
    % Mj índices de columnas

    lM = length(Mn);
    y = 1;
    k = 1;

    while k <= lM && y == 1

        if Mi(k)-Mj(k) > 1 || Mi(k)-Mj(k) < -1
            y = 0;
        else
            k=k+1;
        end
    end
end


b)
function [Tn,Ti,Tj] = extraerTri(Mn,Mi,Mj)
    % Mn valores
    % Mi índices de filas
    % Mj índices de columnas

    lM = length(Mn);

    if lM == 0
        Tn = [];
        Ti = [];
        Tj = [];
    else

        [Tn,Ti,Tj] = extraerTri(Mn(2:lM),Mi(2:lM),Mj(2:lM));

        if Mi(1)-Mj(1) <= 1 && Mi(1)-Mj(1) >= -1
            Tn = [Mn(1), Tn];
            Ti = [Mi(1), Ti];
            Tj = [Mj(1), Tj];
        end
    end
end
```