Examen febrero 2018 - Computación 1

Problema 1

a)

26 / 2

0 13 / 2

1 6 / 2

0 3 / 2

1 1

1 1 0 1 0

$1,101 * 2^4$

desp = $2^3-1 = 15$

exp = 19

19 / 2

1 9 / 2

1 4 / 2

0 2 / 2

0 1

0 10011 1010000

31 / 2

1 15 / 2

1 7 / 2

1 3 / 2

1 1

1 1 1 1 1

$1,1111 * 2^4$

exp = 19

0 10011 1111000

b)

1,1111

1,1010

11,1001

1,11001 * 2^5

0 10100 1100100

c)

111001 bin

39 Hex

16 * 3 + 9 = 57 dec

Problema 2

```
function [F,C] = cantFCRec(As, Ai, Aj)
  s = length(As);
  if s == 1
      F = Ai(1);
      C = Aj(1);
  else
    [F,C] = cantFCRec(As(2:s), Ai(2:s), Aj(2:s));
    if (Ai(1)  > F)
      F = Ai(1);
    end
    if (Aj(1)  > C)
      C = Aj(1);
    end
  end
end
```

```
%denso sin asumir particionamiento
function R = buscarMaxRec(A)
  [m,n] = size(A);
  if m == 1 && n == 1
    R = A(1,1);
  elseif m == 1 || n == 1
      if m == 1
         R = buscarMaxRec(A(1,2:n));
      else
         R = buscarMaxRec(A(2:n,1));
      end
       if A(1,1) > R
          R =  A(1,1);
       end
  else
    Rf = buscarMaxRec(A(1,1:n));
    R = buscarMaxRec(A(2:m,1:n));
    if Rf > R
      R =  Rf;
    end
  end
end
```

```
%disperso
function R = buscarMaxRec(As, Ai, Aj)
  s = length(As);
  if s == 1
    R = As(1);
  else
    R = buscarMaxRec(As(2:s), Ai(2:s), Aj(2:s));
    if As(1) > R
      R =  As(1);
    end
  end
end


function [MaxR, MaxC] = maxpFCIt(As, Ai, Aj, m, n)
  MaxR = zeros(m,1);
  MaxC = zeros(n,1);
  for i = 1:length(As)
    if As(i) > MaxR(Ai(i))
      MaxR(Ai(i)) = As(i);
    end
    if As(i) > MaxC(Aj(i))
      MaxC(Aj(i)) = As(i);
    end
   end
 end
```

Problema 3

```
function R = comprimirIt(T) %
  R = [T(1)];
  cant = 1;
  for i = 2:length(T)
    if T(i-1) == T(i)
      cant = cant + 1;
    else
      R = [R cant T(i)];
      cant = 1;
    end
  end
  R = [R cant];
end

function R = descomprimirRec(T) %
  if length(T) == 2
    R = ones(1, T(2)) * T(1);
  else
    R = descomprimirRec(T(3:length(T)));
    R = [ones(1, T(2)) * T(1) R];
  end
end
```

Problema 4

```
function R = cuantoEstaIt(T,X)
  R = 0;
  for i = 1:length(T)
    if T(i) == X
      R = R+1;
    end
  end
end

function R = estaXNIt(T,X,n)
  cant = 0;
  i = 1;
  R = 0;
  while (cant < n) && (i <= length(T))
    if T(i) == X
      cant = cant+1;
    end
    i = i+1;
  end
  if cant >= n
    R = 1;
  end
end

function R = estaXNRec(T,X,n)
  l = length(T);
  if n == 0
      R = 1;
  elseif l == 0
      R = 0;
  elseif T(1) == X
       R= estaXNRec(T(2:l),X,n-1);
  else
      R= estaXNRec(T(2:l),X,n);
end
```