Examen Diciembre - 8 de Diciembre de 2012

Problema 1	17 (2,2,2,2,3,3,3) ptos	

a) Un mismo programa en un lenguaje que se compila se ejecutará más rápido que escrito en otro que se interpreta. Los interpretados en general son más fáciles de portar a otras plataformas.

b) Las variables en los scripts son globales, mientras que en las funciones son locales, es decir que solamente existen mientras se ejecuta la función y solamente la función la conoce y puede manipular.

c) Ventajas de la Recursión: Soluciones simples y claras a problemas inherentemente recursivos.
Desventajas de la Recursión:
- Ineficiencia
- Requieren más memoria (nuevo espacio de memoria en cada llamada recursiva)
- Redundancia en cálculos

d) 1 10000101 11010101000000000000000

s = -1
exp = 133-127 = 6
1.f = 1.11010101000000000000000...
muevo la coma 6 lugares=> 1110101,01=117,25 => el valor que representa es -117.25

e) 31,625

0 10000011 11111010000000000000000

f) 1 00000000 00000001000000000000000
-2^{-134}

g) -111
10010001

Problema 2	20 (6, 4, 4, 6) ptos	

a)
```
function prox=proxEstado(v,pos,L)
  n=length(v);
  maxInd=pos+L;
  minInd=pos-L;
  if  minInd<1
     minInd=1;
  end
  if  maxInd>n
    maxInd=n;
  end
  suma=0;
  for i=minInd:maxInd
     suma = suma+ v(i);
  end
  cantCeldas = maxInd-minInd + 1;
  cant = floor(cantCeldas/2);
  if  suma > cant
      prox=1;
  else
      prox=0;
  end
```

b)
```
function proxV=proxConfiguracion(v,L)
   n=length(v);
   proxV = [];
   for pos=1:n
       proxV = [proxV, proxEstado(v,pos,L)];
   end
```

c)
```
function ret = iguales(v1,v2)
   n1=length(v1);
   n2=length(v2);
   if n1~=n2
       ret=0;
   else
       ret=1;
       i=1;
       while i<=n1 & ret
          if(v1(i)~=v2(i))
              ret=0;
          end
          i=i+1;
       end
   end
```

d)
```
function ret=cuantosPasos(v1,v2,L)
   ret=0;
   while iguales(v1,v2)==0
      v1=proxConfiguracion(v1,L);
      ret=ret+1;
   end
```

Problema 3	24 (8, 8, 8) ptos	

a)
```
function vcomp=compactar(v)
     n=length(v);
     if n==0
         vcomp=[];
     else
        vcomp=compactar(v(1:n-1));
        if v(n)==1
          vcomp=[vcomp n];
        end
     end
```

b)
```
function or=orCompactado(c1,c2)
    n1=length(c1);
    n2=length(c2);
    if n1==0
      or = c2;
    elseif n2==0
        or = c1;
    elseif c1(1)==c2(1)
        or = [c1(1) orCompactado(c1(2:n1),c2(2:n2))]
    elseif c1(1)>c2(1)
        or = [c2(1), orCompactado(c1,c2(2:n2))]
    else
        or = [c1(1), orCompactado(c1(2:n1),c2]
    end
```

c)
```
function ret=saltos(v)
   n=length(v);
   if n==0 | n==1
      ret=[];
   else
       ret=[saltos(v(1:n-1))];
       if(v(n)~=v(n-1))
         ret = [ret, n-1];
       end
   end
```

Problema 4	20 (10, 10) ptos	

a)
```
Alternativa 1: Sin la multiplicación de escalar por vector de matlab
function ret=multiplicar(v, M)
   [m,n]=size(M);
   if(m==0)
     ret=[];
   else
      prod=0;
      for j=1:n
          prod=v(i)*M(1,j) + prod;
      end
      ret=[prod; multiplicar(v, M(2:m,:)) ]
   end

Alternativa 2: Con la multiplicación de escalar por vector de matlab
function ret=multiplicar(v, M)
  [m,n]=size(M);
  if(m==0)
     ret=[];
  else
    prod=v(1)*M(:,1);
    ret= prod + multiplicar(v(2:n), M(:,2:n))
  end
```

b)
```
function ret = multiplicar(v, M)
  [m,n]=size(M);
  ret=[];
  for i=1:m
     prod=0;
     for j=1:n
        prod=v(j)*M(i,,j) + prod;
     end
     ret=[ret; prod];
  end
```

Problema 5	19 (9, 10) ptos	

a)
```
function res=multVectDisp(vc,vfd,vfp)
   n=length(vfd);
   res=0;
   for i=1:n
      res=vc(vfp(i))* vfd(i) + res;
   end
```

b)
```
function ret=multvectMatDisp(v,af,ac,ad,cantFilas)
  lm=length(af);
  ret=zeros(cantFilas,1);
  for i=1:lm
     ret(af(i))=ret(af(i)) + v(ac(i))*ad(i);
  end
```

c)