



Examen - Diciembre de 2003 - Soluciones

Parte 1 30 pts

	1	2	3	4	5	6	7	8	9	10
respuesta	d	a	b	b	c	c	b	c	b	c

Parte 2

Problema 1	ptos	
-------------------	------	--

- a) (50) se representa en complemento a uno como 00110010
Su complemento (o sea, -50) se representa como 11001101

33×2^{-5} se puede escribir como 100001×2^{-5} , luego de normalizado queda $1,00001 \times 2^0$.
Convertimos el exponente 0 a exceso de 8 bits: $0 + 2^{8-1} - 1 = 127$, que traducido a binario queda 01111111.

Finalmente el resultado expresado en punto flotante de simple precisión es:

0 01111111 000010000000000000000000

- b) Procedemos a realizar la suma:

```

111  1
 1011011
  0110010
  0001101
  -----
    1
  0001110

```

Observemos que el bit de carry y el de overflow coinciden, por lo que el resultado no se va de rango.

Pasamos el resultado a decimal y es: 14

- c) La primera cadena representa el entero sin signo 2163212288
La segunda cadena representa el entero sin signo 5505024

1 00000001 111000000000000000000000

El exponente lo pasamos a decimal: $1 - ((2^{8-1}) - 1) = -126$.

Luego podemos escribir: $1,111 \times 2^{-126} = 1111 \times 2^{-129} = -15 \times 2^{-129}$

0 00000000 101010000000000000000000

Este número está desnormalizado, por lo que el exponente es (-126).

Luego podemos escribir: $0,10101 \times 2^{-126} = 10101 \times 2^{-131} = 21 \times 2^{-131}$

Problema 2	pts	
-------------------	-----	--

El cadete

```

function ruta = MensajeroRecurso (calle, puerta)
    if puerta == 0
        ruta = [];
    else
        ruta = [puerta, MensajeroRecurso(calle,calle(puerta))];

```



end

```
function ruta = MensajeroIterativo (calle, puerta)
    ruta = [puerta];
    termine = 0;
    while (~termine)
        if (calle(puerta) == 0)
            termine = 1;
        else
            ruta = [ruta calle(puerta)];
            puerta = calle(puerta);
        end
    end
end
```

Problema 3	pts	
-------------------	-----	--

```
function y=escapicua(num)
% convierto el número a vector
V = []
while num ~= 0
    V = [ mod( num, 10 ) V ]
    num = floor( num / 10 )
end

% analizo si el vector es capicúa
n = length (V);
if (n == 1 | n == 0)
    y = 1;
elseif V(1) == V(n)
    % convierto el vector a número para llamada recursiva
    num = 0;
    for ind = 2:length(V)-1
        num = (num * 10) + V(ind);
    end
    n = length (V);
    y = escapicua(V(2:n-1));
else
    y = 0;
end
```