



Examen - 10 de diciembre de 2011 - Solución

Problema 1 – 25 puntos (3, 8, 8, 6)

a) La función *desconocida1* tiene tres errores, que se corrigen a continuación:

```
01 % calcula un máximo raro.
02 % recibe un vector de enteros no
03 % negativos y un entero mayor o
04 % igual a uno.
05
06 function max = desconocida1(v, k)
07 n = length(v);
08 i = 0; i = 1;
09 a = -1;
10 while i <= n
11     if mod(i, k) == 0 & a < v(i)
12         a = v(i);
13     end
14     i = i+1;
15 end
16 max = a;
```

b)

```
01 function max = desconocida1(v, k)
02 n = length(v);
03 if n == 0
04     max = -1;
05 else
06     max = desconocida1(v(1:n-1), k);
07     if mod(n, k) == 0 & v(n) > max
08         max = v(n);
09     end
10 end
```

c)

```
01 function v = desconocida2(w, dir)
02 v = [];
03 n = length(w);
04 if n ~= 0
05     izq = 1;
06     der = n;
07     for i = 1:n
08         if dir == 1
09             v(i) = w(izq);
10             if w(izq) == 0
11                 dir = -1;
12             end
13             izq = izq + 1;
14         else
15             v(i) = w(der);
16             if w(der) == 0
17                 dir = 1;
18             end
19             der = der - 1;
20         end
21     end
22 end
```



d)

Versión corta:

$$A = \begin{array}{|c|c|c|} \hline 0 & 00010100 & 010100000000000000000000 \\ \hline s & \text{exp} & f \\ \hline \end{array}$$

$$B = \begin{array}{|c|c|c|} \hline 0 & 00010100 & 110100000000000000000000 \\ \hline s & \text{exp} & f \\ \hline \end{array}$$

$$C = \begin{array}{|c|c|c|} \hline 0 & 10000010 & 110000010000000000000000 \\ \hline s & \text{exp} & f \\ \hline \end{array}$$

A y B tienen el mismo exponente, así que no es necesario alinear las mantisas. Por lo tanto, sumamos las mantisas directamente y luego normalizamos el resultado:

$$\begin{array}{r} 1, 0101 \\ + 1, 1101 \\ \hline 11, 0010 \end{array}$$

$$A + B = \begin{array}{|c|c|c|} \hline 0 & 00010101 & 100100000000000000000000 \\ \hline s & \text{exp} & f \\ \hline \end{array}$$

Para calcular la suma de A y C, es necesario alinear las mantisas, desplazando a la derecha la mantisa que tiene el exponente más pequeño tantos lugares como la diferencia entre los exponentes. Como el exponente de A vale -107 y el de C vale 3 (luego de calculado el desplazamiento), vemos que su diferencia (104) es mayor que la cantidad de bits de la mantisa (23). Por lo tanto, en la suma el valor de A se "pierde" y el resultado es igual a C:

$$A + C = \begin{array}{|c|c|c|} \hline 0 & 10000010 & 110000010000000000000000 \\ \hline s & \text{exp} & f \\ \hline \end{array}$$



Versión larga:

Primero vemos qué representa cada tira de bits:

La primera tira es:

0	00010100	01010000000000000000000000000000
s	exp	f

Representa al número:

$$N_A = +1, f \times 2^e, \text{ donde}$$

$$1, f = 1,0101, \text{ y}$$

$$e = \text{exp} - (2^7 - 1) \\ = \text{exp} - 127$$

Como  $\text{exp} = 10100_2 = 20_{10}$ , entonces:

$$e = 20 - 127 = -107$$

Por lo tanto, el número que estamos buscando es:

$$N_A = +1,0101 \times 2^{-107}$$

Aplicando el mismo procedimiento, vemos que:

$$N_B = +1,1101 \times 2^{-107} \quad \text{y} \quad N_C = +1,11000001 \times 2^3$$

**Cálculo de las sumas**

Para la suma de  $N_A + N_B$  no hay que alinear las mantisas porque los exponentes son iguales. Por lo tanto, sumamos las mantisas directamente y luego normalizamos el resultado:

$$\begin{array}{r} 1, 0101 \\ + 1, 1101 \\ \hline 11, 0010 \end{array}$$

Por lo tanto,

$$N_A + N_B = 1,1001 \times 2^{-106}, \text{ representado por la tira:}$$

0	00010101	10010000000000000000000000000000
---	----------	----------------------------------

Para  $N_A + N_C$  es necesario alinear las mantisas, desplazando a la derecha la mantisa que tiene el exponente más pequeño tantos lugares como la diferencia entre los exponentes. Como la diferencia entre los exponentes (104) es mayor que la cantidad de bits de la mantisa (23), el valor de  $N_A$  se “pierde”. El resultado, por lo tanto, es igual a  $N_C$ .



**Problema 2 – 30 puntos (18, 12)**

a)

```
01 function [As, Ai, Aj] = filtrarColor(M, min, max)
02 % Asumo que min >= 0 y max >= 0
03 As = [];
04 Ai = [];
05 Aj = [];
06 [m, n] = size(M);
07 for i = 1:m
08     for j = 1:n
09         if M(i,j) > min & M(i,j) < max
10             As = [As M(i,j)];
11             Ai = [Ai i];
12             Aj = [Aj j];
13         end
14     end
15 end
```

b) la función recursiva *mezclarImágenes* que dadas dos fotografías de igual tamaño, representadas como matrices **completas** de enteros, retorna una fotografía en que cada píxel de la misma se corresponde con el color de mayor intensidad (el entero mayor) de las fotografías originales.

```
01 function C = mezclarImágenes(A, B)
02 n = size(A, 1);
03 if n == 1
04     if A > B
05         C = A;
06     else
07         C = B;
08     end
09 else
10     medio = n/2;
11     C1 = mezclarImágenes(A(1:medio, 1:medio), B(1:medio, 1:medio));
12     C2 = mezclarImágenes(A(1:medio, medio+1:n), B(1:medio, medio+1:n));
13     C3 = mezclarImágenes(A(medio+1:n, 1:medio), B(medio+1:n, 1:medio));
14     C4 = mezclarImágenes(A(medio+1:n, medio+1:n), B(medio+1:n, medio+1:n));
15     C = [C1 C2; C3 C4];
16 end
```



**Problema 3 – 30 puntos (10, 20)**

a)

```
01 function derpol = dervpol_rec(pol, x)
02 grado = length(pol)-1;
03 if grado == 0
04     derpol = 0;
05 else
06     derpol = grado*pol(1)*x^(grado-1) + dervpol_rec(pol(2:grado+1), x);
07 end
```

b)

```
01 function [x1, conv] = NR(pol, x0, tol, maxiter)
02 derpol = dervpol_rec(pol, x0);
03 x1 = -polyval(pol, x0) / derpol + x0;
04 s = sign(derpol);
05 conv = 1;
06 i = 1;
07 while abs(x1-x0) >= tol & i <= maxiter & conv
08     x0 = x1;
09     derpol = dervpol_rec(pol, x0);
10     x1 = -polyval(pol, x0) / derpol + x0;
11     if sign(derpol) ~= s
12         conv = 0;
13     end
14     i = i + 1;
15 end
```



**Problema 4 – 15 puntos (5, 10)**

a)

```
01 function v = NumToVec(n)
02 r = mod(n, 10);
03 d = floor(n/10);
04 if d == 0
05     v = [r];
06 else
07     v = [NumToVec(d) r];
08 end
```

b)

```
01 function es = esPalindromo(n)
02 v = NumToVec(n);
03 n = length(v);
04 es = 1;
05 i = 1;
06 medio = floor(n/2);
07 while i <= medio & es
08     if v(i) ~= v(n-i+1)
09         es = 0;
10     end
11     i = i + 1;
12 end
```