

**Examen - Febrero de 2002 - Soluciones**

<b>Parte 1</b>	30 pts	
----------------	--------	--

Pregunta	1	2	3	4	5	6	7	8	9	10
Casilla de control										
<b>1</b>	<b>b</b>	<b>c</b>	<b>c</b>	<b>a</b>	<b>c</b>	<b>b</b>	<b>d</b>	<b>b</b>	<b>b</b>	<b>b</b>
<b>2</b>	<b>c</b>	<b>b</b>	<b>b</b>	<b>a</b>	<b>b</b>	<b>d</b>	<b>b</b>	<b>b</b>	<b>c</b>	<b>b</b>
<b>3</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>b</b>	<b>c</b>	<b>c</b>	<b>b</b>	<b>b</b>	<b>b</b>	<b>d</b>

<b>Parte 2</b>	70 pts	
----------------	--------	--

<b>Problema 1</b>	30 pts	
-------------------	--------	--

a) i)

```
% realiza el producto de 2 polinomios
function y=pprod(a,b)
y=zeros(length(a)+length(b)-1,1); % creo vector resultado con 0
for i=0:length(a)-1
    for j=0:length(b)-1
        y(i+j+1)=y(i+j+1)+a(i+1)*b(j+1); %calculo y agrupo términos en la solución
    end
end
```

ii)

```
% funcion que realiza la derivada de un polinomio
function y=pder(a)
for i=1:length(a)-1
    y(i)=a(i+1)*i;
end
```

b) i)

```
% suma fracciones de polinomios a/b + c/d = e/f
function [e,f]=fsum(a,b,c,d)
% e=a*d+c*b
% f=b*d
e1=pprod(a,d); e1=e1(:);
e2=pprod(c,b); e2=e2(:);
% calculo cada termino de la suma, armamos vectores de igual tamaño
% para poder aplicar el + de Matlab sin que existan problemas de dimensiones
e=[e1; zeros(length(e2)-length(e1),1)] + [e2; zeros(length(e1)-length(e2),1)];
f=pprod(b,d);
```

ii)

```
% deriva una fraccion de polinomios
function [c,d]=fder(a,b)
% c=a*b - a*b'
% d=b^2=b*b
ap=pder(a); %calculo las derivadas de los polinomios
bp=pder(b);
c1=pprod(ap,b); c1=c1(:);
c2=pprod(a,bp); c2=c2(:);
c=[c1; zeros(length(c2)-length(c1),1)] - [c2; zeros(length(c1)-length(c2),1)];
```



d=pprod(b,b);

<b>Problema 2 (curso 2001)</b>	15 ptos	
--------------------------------	---------	--

a)  $123 \times 2^{-5} \rightarrow 1111011 \times 2^{-5} \rightarrow 1,111011 \times 2^1$   
 s 0  
 e  $127 + 1 = 10000000$   
 m 111011000000000000000000  
  
 0 10000000 111011000000000000000000

b) 0 10001001 111100000000000000000000  
  
 exp =  $137 - 127 = 10$

$1,1111 \times 2^{10} \rightarrow 11111 \times 2^6 \rightarrow 31 \times 2^6$

c) +inf  $\rightarrow$  0 11111111 000000000000000000000000  
 -inf  $\rightarrow$  1 11111111 000000000000000000000000

- d)
- I. desnormalizado
  - II. normalizado
  - III. NaN
  - IV. NaN
  - V. desnormalizado

<b>Problema 2 (cursos anteriores a 2001)</b>	15 ptos	
--	---------	--

1. En punto flotante (simple precisión)  $1s\ 8e\ 23m$  (  $s^*1, mx2e$  )
2.  $\epsilon_{MACH} = \min(x : FP(1+x) > 1)$
3. Se restan números grandes pero “relativamente parecidos”, de tal forma que el resultado tiene un error relativo grande.
- 4
- (a) Shift Out, se presenta al sumar dos números en los cuales uno es relativamente pequeño al otro.
- (b)
  - 1) Guardar los valores en un vector y realizar la sumatoria de el valor más chico al más grande.
  - 2) Tomar de a 10 valores sumarlos y guardarlos en variable y luego sumar dichas variables.

<b>Problema 3</b>	25 ptos	
-------------------	---------	--

a)

function c = invertir (b)

l = length (b);  
 if l == 1

```
c = b;  
else  
  c = [invertir(b(2:l)), b(1)];  
end
```

**b)**

```
function bin = Octal(num)  
k = 1;  
while num > 0  
  resto = rem(num, 8);  
  num = floor( num / 8);  
  c(k) = resto;  
  k = k + 1;  
end  
bin = invertir( c );
```

**c)**

```
function dec = decimal(oct)  
  
l = length (oct)  
if l == 1  
  dec = bin;  
else  
  dec = decimal(oct(2:l));  
  dec = oct(1) * 8^(l-1) + dec  
end
```

Instituto de Computación - Facultad de Ingeniería - Universidad de la R. O. del Uruguay