

Examen - Agosto de 2002 - Soluciones

| | | | | | | | | | | | |
|---------------------------|-----------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| Preguntas | 30 pts | | | | | | | | | | |
| | Pregunta | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Respuesta correcta | | c | d | d | a | a | b | c | b | a | c |

| | | |
|-------------------|---------------|--|
| Problema 1 | 30 pts | |
|-------------------|---------------|--|

```
% busca e imprime todas las ocurrencias del vector tira dentro de la matriz sopa
% si la tira es de un solo elemento, se listan solo las posiciones donde ocurre
```

```
function sopaDeLetras( sopa, tira )
```

```
    [ m, n ] = size( sopa );
    [ p, q ] = size( tira );
```

```
    if ( p == 1 & q == 1 )           % es un unico elemento, busqueda simple
        for i = 1:m
            for j = 1:n
                if ( tira(1,1) == sopa(i,j) )
                    disp('Ocurrencia encontrada en');
                    fila = i
                    columna = j
                end
            end
        end
    end
```

```
    else                               % mas de uno, hay direccion y sentido
        for i = 1:m
            for j = 1:n
```

```
                %verificar verticalmente
                if ( prefijo( tira, obtenerVectorArriba( sopa, i, j ) ) )
                    disp('Ocurrencia encontrada : comienza en');
                    fila = i
                    columna = j
                    disp('leyendo hacia arriba');
                end;
```

```
                if ( prefijo( tira, obtenerVectorAbajo( sopa, i, j ) ) )
                    disp('Ocurrencia encontrada : comienza en');
                    fila = i
                    columna = j
                    disp('leyendo hacia abajo');
                end
```

```
                %horizontalmente
                if ( prefijo( tira, obtenerVectorDerecha( sopa, i, j ) ) )
                    disp('Ocurrencia encontrada : comienza en');
                    fila = i
                    columna = j
                    disp('leyendo hacia la derecha');
                end;
```

```
                if ( prefijo( tira, obtenerVectorIzquierda( sopa, i, j ) ) )
                    disp('Ocurrencia encontrada : comienza en');
                    fila = i
                    columna = j
                    disp('leyendo hacia la izquierda');
                end
```

```
                %en diagonales
```



```

        if ( prefijo( tira, obtenerVectorArribaALaDerecha( sopa, i, j ) ) )
            disp('Ocurrencia encontrada : comienza en');
            fila = i
            columna = j
            disp('leyendo hacia arriba y a la derecha');
        end

        if ( prefijo( tira, obtenerVectorArribaALaIzquierda( sopa, i, j ) ) )
            disp('Ocurrencia encontrada : comienza en');
            fila = i
            columna = j
            disp('leyendo hacia arriba y a la izquierda');
        end

        if ( prefijo( tira, obtenerVectorAbajoALaDerecha( sopa, i, j ) ) )
            disp('Ocurrencia encontrada : comienza en');
            fila = i
            columna = j
            disp('leyendo hacia la abajo y a la derecha');
        end

        if ( prefijo( tira, obtenerVectorAbajoALaIzquierda( sopa, i, j ) ) )
            disp('Ocurrencia encontrada : comienza en');
            fila = i
            columna = j
            disp('leyendo hacia abajo y a la izquierda');
        end

    end

end
end
end

%% da true sii w es un subvector de v, comenzando desde el primer elemento
function res = prefijo ( w, v )

    [ aux, tamW ]= size( w );
    [ aux, tamV ] = size( v );

    if ( tamW > tamV )
        res = 0;
    else

        res = prod( w == v(1:tamW) );

    %     OTRA FORMA :

    %     res = 1;
    %     k = 1;

    %     while ( res & k <= tamW )
    %         res = ( w(k) == v(k) );
    %         k = k + 1;
    %     end

end

end

%funciones para obtener un vector con elementos de una matriz,
%moviendose en cierto sentido a partir de una posicion dada

function v = obtenerVectorArriba ( M, i, j )
    v = ( M(i:-1:1,j) )';
end

function v = obtenerVectorAbajo ( M, i, j )
    [f,c] = size(M);
    v = ( M(i:1:f,j) )';
end
end

```

```
function v = obtenerVectorDerecha ( M, i, j )
    [f,c] = size(M);
    v = M(i,j:1:c);
end

function v = obtenerVectorIzquierda ( M, i, j )
    v = M(i,j:-1:1);
end

function v = obtenerVectorArribaALaDerecha ( M, i, j )

    [f,c] = size(M);
    k = 1;
    m = i;
    n = j;

    while ( m >= 1 & n <= c )

        v(k) = M(m,n);
        k = k + 1;

        m = m - 1;
        n = n + 1;
    end
end

function v = obtenerVectorArribaALaIzquierda ( M, i, j )

    k = 1;
    m = i;
    n = j;

    while ( m >= 1 & n >= 1 )

        v(k) = M(m,n);
        k = k + 1;

        m = m - 1;
        n = n - 1;
    end
end

function v = obtenerVectorAbajoALaDerecha ( M, i, j )

    k = 1;
    m = i;
    n = j;

    [f,c] = size(M);

    while ( m <= f & n <= c )

        v(k) = M(m,n);
        k = k + 1;

        m = m + 1;
        n = n + 1;
    end
end

function v = obtenerVectorAbajoALaIzquierda ( M, i, j )

    k = 1;
    m = i;
    n = j;

    [f,c] = size(M);

    while ( m <= f & n >= 1 )

        v(k) = M(m,n);
        k = k + 1;
```

```

m = m + 1;
n = n - 1;
end
end

```

| | | |
|--------------------------------|---------------|--|
| Problema 2 (curso 2001) | 15 pts | |
|--------------------------------|---------------|--|

a) 0 10000001 111110000000000000000000

b) $119 * 2^{-112}$

c)

NaN 0 11111111 000000000000000000000001

Inf 1 11111111 000000000000000000000000

0 0 00000000 000000000000000000000000

d)

- | | | | | |
|------|---|----------|-------------------------|----------------|
| i. | 1 | 10000001 | 11001101100111000011101 | Normalizado |
| ii. | 1 | 00000000 | 10010001000000010000000 | Desnormalizado |
| iii. | 0 | 00000000 | 11111111111111111111110 | Desnormalizado |
| iv. | 0 | 11111110 | 00000000000000000000000 | Normalizado |
| v. | 0 | 00000000 | 00000000000000000000001 | Desnormalizado |

| | | |
|--|---------------|----------------|
| Problema 2 (cursos anteriores a 2001) | 15 pts | 2, 4, 9 |
|--|---------------|----------------|

a) $\epsilon_{\text{MACH}} = \min(x : \text{FP}(1+x) > 1)$ Por más detalles ver teórico

b)

```

x := 1
While 1+x > 1
  x := x/2
End
x*2

```

c) Ver teórico

Que están previstos:

Not a number: resultado de 0/0

Overflow: número demasiado grande para poder ser representado

$|X| > \max|z|$ $z \in \text{FP}$ Se sustituye por *inf* con signo

Underflow: número demasiado pequeño para poder ser representado.

$x \neq 0$; $|X| < \min|z|$ $z \in \text{FP}, z \neq 0$ Se sustituye por 0

No son detectados por el sistema:**Catastrophic Cancellation:**

Se restan números grandes pero “relativamente parecidos” de tal forma que el resultado tiene un error relativo grande.

Solución: realizar las operaciones en otro orden o realizar otras operaciones “mejor condicionadas”

Shift Out

Se suman dos números: uno relativamente chico con respecto al otro.

Solución: si se están sumando muchos números (por ejemplo sumatorias) realizar las operaciones de forma que primero se sumen los “números chicos”

| | | |
|-------------------|----------------|--|
| Problema 3 | 25 ptos | |
|-------------------|----------------|--|

a)

```
function indice = ind_val_max( v, m )
indice = 1;
maximo = v(1);
for i = 2:m
    % en este caso si hay varios max me quedo
    % con el primero
    if v(i) > maximo
        maximo = v(i);
        indice = i;
    end
end
```

b)

```
function v = intercambiar( v, i, j)
aux = v(i);
v(i) = v(j);
v(j) = aux
```

c)

```
function v = ordenar( v )
for i = length(v):-1:2
    v = intercambiar( v, ind_val_max(v, i), i);
end
```