

Primer Parcial – 26 de setiembre de 2024

Para la resolución de los diferentes ejercicios **solamente** podrá utilizar las siguientes funciones brindadas por **Octave**:

- `length()` y `size()`
- `mod()`, `div()` y `rem()`
- `floor()`, `ceil()` y `round()`
- `zeros()` y `ones()`

**Notas:** - En todos los ejercicios se deben usar las estructuras de control adecuadas para cada caso. Por ejemplo: se controlará el uso correcto de `for` y `while`.  
- No se deben realizar más iteraciones que las necesarias para resolver los problemas.

<b>Problema 1</b>	4 (2, 2) pts	
-------------------	--------------	--

a)  $n1 \leq n3$

b)  $n2 \leq n3$

<b>Problema 2</b>	4 pts	
-------------------	-------	--

a = 11, b = 3, c = indefinida

<b>Problema 3</b>	8 pts	
-------------------	-------	--

```
function v = cambiarUnosCeros(v)
    for i=1:length(v)
        if v(i)==0
            v(i)=1;
        elseif v(i)==1
            v(i)=0;
        endif
    endfor
endfunction
```

<b>Problema 4</b>	8 pts	
-------------------	-------	--

```
function primer=primeroMayorQueX(v,x)
    lv=length(v);
    i=1;
    primer=-1;
    while i<=lv && v(i)<=x
        i=i+1;
    endwhile
    if i<=lv
        primer=v(i);
    endif
endfunction
```

Otra opción:

```
function primer=primeroMayorQueX(v,x)
    lv=length(v);
    i=1;
    primer=-1;
    while i<=lv && primer===-1
        if v(i)>x
            primer=v(i);
        else
            i=i+1;
        endif
    endwhile
```

endfunction

<b>Problema 5</b>	8 pts
-------------------	-------

Este ejercicio es similar al Ejercicio 3 que se resolvió en la clase 10 de teórico.

```
function vOrd=diferencia(v1,v2)
    lv1=length(v1);
    lv2=length(v2);
    vOrd=[];
    i1=1;
    i2=1;
    while i1<=lv1 && i2<=lv2
        if v1(i1)==v2(i2)
            i1=i1+1;
            i2=i2+1;
        elseif v1(i1)<v2(i2)
            vOrd=[vOrd v1(i1)];
            i1=i1+1;
        else
            i2=i2+1;
        endif
    endwhile

    % si i1 es mayor que lv1 no agrega nada.
    for i=i1:lv1
        vOrd=[vOrd v1(i)];
    endfor
    % Equivalentemente al for se podría haber hecho vOrd=[vOrd v1(i1:lv1)];
endfunction
```

<b>Problema 6</b>	8 pts
-------------------	-------

Este ejercicio tiene puntos de contacto con la función `separarPalabras` de la primera tarea de programación del laboratorio.

```
function [posUnos, posCeros] = posSecUnosCeros(v)
    posUnos = [];
    posCeros = [];
    ult = -1;
    for i=1:length(v)
        if ult~=1 && v(i)
            posUnos=[posUnos i];
        elseif ult~=0 && ~v(i)
            posCeros=[posCeros i];
        endif
        ult=v(i);
    endfor
endfunction
```

Otra opción:

```
function [posUnos, posCeros] = posSecUnosCeros(v)
    posUnos = [];
    posCeros = [];
    i = 1;
    n = length(v);
    while i <= n
        if v(i)
            posUnos = [posUnos, i];
            while i <= n && v(i)
                i = i + 1;
            endwhile
        else
            posCeros = [posCeros, i];
            while i <= n && ~v(i)
                i = i + 1;
            endwhile
        end
    endwhile
endfunction
```

COMPUTACIÓN 1  
Instituto de Computación

```
        endwhile  
    else  
        posCeros = [posCeros, i];  
        while i <= n && ~v(i)  
            i = i + 1;  
        endwhile  
    endif  
endfunction
```