

COMPUTACIÓN 1

Instituto de Computación

Examen - 19 de febrero de 2020



- Duración del examen: 3 hs.
- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). Apague su celular.
- **Sólo** se contestarán preguntas sobre interpretación de la letra hasta 30 minutos antes de la finalización del mismo.
- Las partes no legibles del parcial se considerarán no escritas
- En la primer hoja a entregar ponga con LETRA CLARA, en el ángulo superior derecho, su nombre, número de cédula de identidad y cantidad de hojas -en ese orden-; las demás hojas es suficiente con nombre, número de cédula y número de página.

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

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

| | | |
|-------------------|--------------------|--|
| Problema 1 | 15 pts (3,3,3,3,3) | |
|-------------------|--------------------|--|

a) $[101000]_2 = 1 \times 2^5 + 1 \times 2^3 = 32 + 8 = \mathbf{40}$

b) $[1000111010111]_2 = 001\ 000\ 111\ 010\ 111 = \mathbf{[10727]_8}$

c) **01001**

d) Exponente en representación con desplazamiento: $00000100_2 = 4_{10}$

M = 127, entonces el exponente vale $4 - 127 = -123$

Entonces el número es $1,111_2 * 2^{-123} = 1111_2 * 2^{-126} = \mathbf{15 * 2^{-126}}$

e) $3,25 = 11,01_2 = 1,101_2 * 2^1$

Como M = 127, el exponente es $1 + 127 = 128 = 10000000_2$

Entonces: **0 10000000 101000000000000000000000**

| | | |
|-------------------|---------------|--|
| Problema 2 | 16 ptos (8,8) | |
|-------------------|---------------|--|

a)

```
function esRaiz = EsRaiz(p,X)

    n = length(p);
    suma = 0;
    for i = 0 : n-1
        suma = suma + p(n-i) * X^i;
    end
    if suma == 0
        esRaiz = 1;
    else
        esRaiz = 0;
    end
end
```

b)

```
function s = SumaPol(p,q)
    lp = length(p);
    lq = length(q);

    if lp == 0 || lq == 0
        s = [p,q];
    else
        s = SumaPol(p(1 : lp-1), q(1 : lq-1));
        if length(s)>0 || (length(s)==0 && p(lp) + q(lq)~=0)
            s = [s, p(lp) + q(lq)];
        end
    end
end
```

| | | |
|-------------------|------------------|--|
| Problema 3 | 24 ptos (12, 12) | |
|-------------------|------------------|--|

a)

```
function [Bv, Bf, Bc] = CambioA(Av, Af, Ac)
    n = length(Av);
    if n == 0
        Bv = [];
        Bf = [];
        Bc = [];
    else
        [Bv, Bf, Bc] = CambioA(Av(2:n), Af(2:n), Ac(2:n));
        if mod(Af(1) + Ac(1), 2) == 0
            Bv = [Av(1) * 3, Bv];
        else
            Bv = [Av(1), Bv];
        end
        Bf = [Af(1), Bf];
    end
```

```
Bc = [Ac(1), Bc];
end
end
```

```
b)
function p = Pertenece(Av, Af, Ac, X)
    p = 0;
    i = 1;
    while i <= length(Av) && p == 0
        if Av(i) == X
            p = 1;
        end
        i = i + 1;
    end
end
```

| | | |
|-------------------|-------------------|--|
| Problema 4 | 31 ptos (6,13,12) | |
|-------------------|-------------------|--|

```
a)
function max = Maximo(v)
    n = length(v);
    if n == 1
        max = v(1);
    else
        max = Maximo(v(2:n));
        if v(1) > max
            max = v(1);
        end
    end
end
```

```
b)
function y = Mayores(M)
    n = size(M,1);

    if n == 2
        y(1) = M(1,2);
        y(2) = M(2,1);
    else
        y = Mayores(M(2:n, 2:n));
        mayorEncimaaux = Maximo(M(1, 2:n));
        mayorDebajoaux = Maximo(M(2:n, 1));
        if mayorEncimaaux > y(1)
            y(1) = mayorEncima2;
        end
        if mayorDebajoaux > y(2)
            y(2) = mayorDebajo2;
        end
    end
end
```

c)

```
function m = MinMax(M)
    [fil, col] = size(M);
    m = [];

    for i = 1:fil
        max = M(i, 1);
        min = M(i, 1);

        for j = 1:col
            if M(i, j) > max
                max = M(i, j);
            elseif M(i, j) < min
                min = M(i, j);
            end
        end

        m = [m ; min max];
    end
end
```

| | | |
|-------------------|---------|--|
| Problema 5 | 14 ptos | |
|-------------------|---------|--|

```
function masPares = masParesRec(v)
    [pares, impares] = cantParesImpares(v);
    masPares = pares > impares;
end

function [pares, impares] = cantParesImpares(v)
    n = length(v);

    if n == 0
        pares = 0;
        impares = 0;
    else
        [pares, impares] = cantParesImpares(v(2:n));
        if mod(v(1), 2) == 0
            pares = pares + 1;
        else
            impares = impares + 1;
        end
    end
end
```