

COMPUTACIÓN 1

Instituto de Computación

Examen – 31 de julio 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()` y `sqrt()`
- ★ `zeros()` y `ones()`

Nota: En todos los ejercicios se deben usar las estructuras de control adecuadas para cada caso.

Problema 1	17 Ptos (2,3,4,4,4)	
-------------------	---------------------	--

- Represente el binario 1101111_2 en notación octal (base 8).
- Represente el octal 1101111_8 en notación hexadecimal (base 16).
- Represente en notación complemento a 2 con 5 bits los números -15 y -16.
- Represente el número 905 en notación punto flotante (1 bit de signo, 5 bits de exponente y 8 de mantisa).
- Plantee y realice la suma de $48 + 905$ en punto flotante (1 bit de signo, 5 bits de exponente y 8 de mantisa).

Nota: Justificar todas las respuestas.

Problema 2	19 ptos (7,4,8)	
-------------------	-----------------	--

Un número es primo cuando sólo es divisible por si mismo y por la unidad.

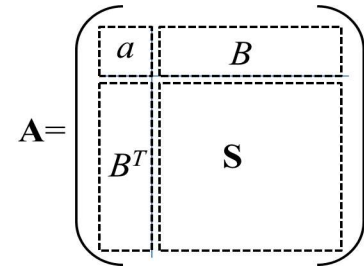
- Escriba la **función iterativa** `r=esprimo(n)` que dado un entero positivo n , devuelve 1 si es primo y 0 en caso contrario. Ayuda: alcanza con probar que n **no** es divisible por ningún entero mayor a 1 y menor o igual a \sqrt{n} .
- Escriba la **función iterativa** `p=primos(n)` que dado un entero positivo dado n , retorna un vector con todos los primos menores o iguales a n .
Atención: Puede utilizar la función de la parte a), aunque no la haya resuelto.
- Escriba una versión **iterativa** de la parte b) que llamaremos `p=Criba(n)`, que cumple el siguiente pseudocódigo:
 - 1) Defina un vector v de largo n con ceros.
 - 2) Desde $i = 2$ hasta $\text{floor}(\sqrt{n})$ haga:
Si $v(i)$ tiene 0 entonces ponga 1 en $v(i*j)$ con j variando entre i y $\text{floor}(n/i)$
 - 3) Ponga en p todos los valores de i tales que $i > 1$ y $v(i) == 0$.

COMPUTACIÓN 1

Instituto de Computación

Problema 3	21 ptos (7, 12)	
-------------------	-----------------	--

- a) Escriba la **función recursiva** `r=VecFIguales(v1,v2)` que dados dos vectores fila $v1$ y $v2$ devuelve 1 si son iguales y 0 en otro caso. Los vectores $v1$ y $v2$ podrían ser de distinto largo.
- b) Una matriz $A_{N \times N}$ es simétrica si $A(i,j) == A(j,i)$ para todo i,j . También es simétrica si se puede descomponer de la forma mostrada a la derecha, donde $a=A(1,1)$, $B=A(1,2:N)$, y $S=A(2:N,2:N)$ que también es una matriz simétrica. Realice la **función recursiva** `r=MatSimRec(A)` que tome en cuenta esta última definición y devuelve 1 si es simétrica y 0 en caso contrario. Puede utilizar la función `VecFIguales` aunque no la haya implementado.



Problema 4	20 ptos (8,12)	
-------------------	----------------	--

- a) Escriba la **función recursiva** `n=cant1001s(s)` que dado un vector s con 0's y 1's, cuente la cantidad de veces que la secuencia `[1 0 0 1]` está contenida en el vector.

Ej: `cant1001s([1 0 1 0 0 1 1 0 0 1]) == 2`

`cant1001s([1 1 0 1 0 0 1 0 0 1]) == 2`

`cant1001s([1 1 0 1 0 0 0 0 0 1]) == 0`

- b) Programe la **función iterativa** `c=CantPal(s,p)` que dados dos vectores filas s y p con 0's y 1's, cuente la cantidad de veces que p se encuentra dentro de s . El vector p no es vacío.

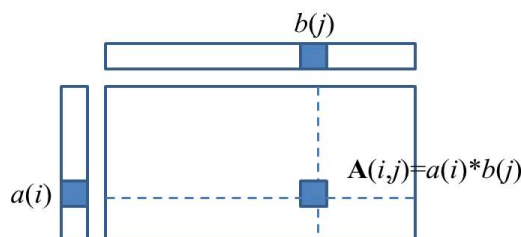
Ej: `CantPal([1 1 0 1 0 0 1 0 0 1], [1 0 0 1]) == 2`

`CantPal([1 0 0 1 0 0 1 0 0 1], [1]) == 4`

`CantPal([1 1 1 1 1], [1 1 1]) == 3`

Problema 5	23 ptos (4,6,13)	
-------------------	------------------	--

- a) Escriba la **función iterativa** `vout=sumNumVecF(x,vin)` que dado un número x y un vector fila vin denso, genera el vector $vout$ también denso, consistente en la suma de x a cada elemento de vin .
- b) Escriba la **función iterativa** `[f,c,v]=prodNumVecF(x,vf)` que dado un número x y un vector fila denso vf , realiza el producto entre x y vf , devolviendo el vector resultante en formato disperso elemental.
- c) El producto $a \cdot b$ entre un vector columna a de largo m y un vector fila b de largo n es una matriz A de dimensión $m \times n$ donde $A(i,j) = a(i) \cdot b(j)$. Implemente `[f,c,v]=prodVecVec(a,b)` como **función recursiva** que dados un vector columna a y un vector fila b en formato denso, devuelve la matriz producto, en formato disperso elemental. Puede utilizar las funciones de las partes a) y b) aunque no las haya implementado.



Nota: Recuerde que una representación de matriz dispersa sólo almacena los valores distintos de 0.