COMPUTACIÓN 1
Instituto de Computación

Examen – 15 de febrero de 2023

- **Duración del examen: 3 Hs.**
- **No** se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). **Apague** su teléfono celular.
- **Escriba las hojas de ambos lados con buena letra**. Las partes no legibles se considerarán no escritas.
- Sólo se contestarán preguntas sobre interpretación de la letra.
- **En la primera hoja, en el ángulo superior derecho, ponga su nombre, número de cédula y cantidad de hojas. En las demás hojas ponga nombre y número de cédula. En todas las páginas ponga 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()`
- `fix(), floor(), ceil() y round()`
- `zeros(), ones() y M'(la transpuesta)`

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 o invocaciones recursivas que las necesarias para resolver el problema

Problema 1	`10 (2,2,2,2,2) ptos`	

En todos los siguientes ejercicios deben justificarse los resultados:

a) Determine *a*, *b* y *c* para que $(2^a-2^b)*2^c$ sea igual al valor decimal de 1111110000_2.

b) Calcule el resultado de 10011 - 00100 en Comp a 1 de 5 bits (indicando si ocurre overflow).

c) Calcule el resultado de 11111 + 00001 en Comp a 2 de 5 bits (indicando si ocurre overflow).

d) Convierta el número 123021_4 a base 2.

e) Obtenga el valor decimal de 110010100000 interpretándolo como un número en punto flotante con 1 bit de signo, 4 bits de exponente y 7 bits de mantisa. El exponente se representa con desplazamiento M=7.

Problema 2	`20 (10,10) ptos`	

a) Escriba en Octave una función ***iterativa*** *acumVec* que dado un vector de números reales *vin* devuelva un vector *vout* tal que *vout(i)* contenga la suma de los elementos de *vin* desde la posición *i* hasta la última posición.

Ejemplo:

con `vin=[1 1 1 1 1 1 1]` devuelve `vout=[7 6 5 4 3 2 1]`

b) Escriba una función ***iterativa*** *corteVec* que dado un vector de números reales positivos devuelva el menor índice del vector tal que la suma de todos los elementos que están a la izquierda de esa posición (sin incluirla) es mayor que la suma de los elementos que están a la derecha. La suma de [] es 0.

Ejemplo:

con `[2 3 5 7 11 13 2] devuelve 5 ya que 2+3+5+7=17 es mayor que 13+2=15`

con `[2 3 5 7 11 13 234] devuelve 7 ya que 2+3+5+7+11+13 es mayor que 0 (suma de [])`

Problema 3	`10 ptos`	

Escriba en Octave una función ***recursiva*** *cortarBilatX* que reciba un vector de enteros *vin* y un número entero *x*, y devuelva un vector *vout* que contenga los elementos de *vin* desde la primera ocurrencia de *x* hasta la última (incluyendo los extremos *x*). Si *x* está solo una vez, el resultado es [*x*], y si no está el resultado es []. Ejemplo:
`con [2,4,3,4,4] y x=4 devuelve [4,3,4,4]; con x=3 devuelve [3]; con x=8 devuelve []`

Problema 4	17 (7,10) ptos.	

a) Escriba en Octave una función ***iterativa*** que reciba una matriz cuadrada dispersa en formato elemental, y devuelva la traza de la matriz, es decir, la suma de los elementos de la diagonal.

b) Escriba una función ***recursiva*** que reciba una matriz no vacía, dispersa, en formato elemental, cuyos elementos se ordenan por filas, y devuelva un vector que en la posición *i* contenga el índice donde comienza la fila *i* en los vectores que componen la matriz dispersa. Se puede asumir que hay al menos un elemento distinto de cero en cada fila.
Ejemplo:

Si la matriz es:

x	x	x	x
y	0	0	0
0	z	0	z
0	0	t	t

la función debe devolver el vector `[1 5 6 8]`

(represente la matriz como dispersa ordenada por filas para comprender mejor el ejemplo).

Problema 5	15 ptos	

Escriba en Octave una función ***recursiva*** *difRec* que reciba dos vectores *v1* y *v2* de enteros ordenados de forma ascendente y devuelva un vector compuesto por los elementos que están en *v1* pero no en *v2* y los elementos que están en *v2* pero no en *v1*.

Ejemplo:

```
difRec([2 3 4],[2 3])      -> [4];       difRec([2 3 4],[2 3 5 10])      -> [4 5 10];
difRec([2 3 4],[])         -> [2 3 4];   difRec([2 3 4],[2 3 4])         -> [];
```

Problema 6	28 (10,8,10) ptos	

a) Una matriz triangular inferior contiene todas las entradas sobre la diagonal iguales a 0. Escriba una función ***iterativa*** *esTriangInf* que reciba una matriz cuadrada M y devuelva 1 si la matriz es triangular inferior y 0 en caso contrario.

Ejemplos:

```
EsTriang([1 0 0                  esTriang([1 0 1
          2 1 0                            0 1 2
          2 0 1])-> 1;                     0 1 0)     -> 0


EsTriang([1 0 0                  esTriang([1 0 1
          0 1 0                            0 1 2
          0 0 1])-> 1;                     0 0 1)     -> 0
```

Nota: para este ejercicio consideramos que las matrices de tamaño 1x1 y [] siempre son triangulares inferiores.

b) Escriba una función ***recursiva*** *esCeroVec* que reciba un vector y devuelva 1 si es vacío o está compuesto completamente por ceros, devolviendo 0 en caso contrario.

c) Escriba una función ***recursiva*** *esTriangInfRec* que reciba una matriz y devuelva 1 si es triangular inferior y 0 en caso contrario. Se sugiere usar la función de la parte anterior.