COMPUTACIÓN 1
Instituto de Computación

2do Parcial -26 de Noviembre de 2010- 2a parte

- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). Apague su teléfono celular.
- **Sólo** se contestarán preguntas sobre interpretación de la letra hasta 20 minutos antes de la finalización del mismo.
- Escriba las hojas de un sólo lado.
- Las partes no legibles del examen 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.

Problema 1 (13 puntos)

Sea la siguiente operación, aplicable a cualquier entero positivo: si el número es par, se divide entre 2; si el número es impar, se multiplica por 3 y se suma 1. La conjetura de Collatz plantea que si se aplica esta operación de forma sucesiva siempre se llega al valor 1, para cualquier valor inicial.

Por ejemplo: $f(6)=3, f(3)=10, f(10)=5, f(5)=16, f(16)=8, f(8)=4, f(4)=2$ y $f(2)=1$.

Se pide implementar la función ***recursiva*** *collatz* en MatLab que, dado un entero positivo ***N***, devuelva un vector con todos los resultados de aplicar la operación hasta llegar a 1. Para controlar que la ejecución termine en algún momento (se trata de una conjetura, por lo que aún no se ha podido demostrar su veracidad o falsedad), la función debe recibir como segundo parámetro un contador que se decrementa con cada aplicación de la operación. Al llegar el contador a cero, la función debe devolver el vector con los resultados calculados hasta el momento. Ejemplos:

```
>> collatz(6, 100)              >> collatz(6, 5)
ans = [6 3 10 5 16 8 4 2 1]     ans = [6 3 10 5 16]
```

Problema 2 (6, 6 y 5 puntos, respectivamente)

Implementar en Matlab:

a) la función ***iterativa*** *binarioADecimal*, que recibe un número binario sin signo, representado como un vector de bits (ceros y unos), y devuelve el decimal correspondiente. Ejemplo:

```
>> binarioADecimal([1,0,0,1])
ans = 9
```

b) la función ***iterativa*** *excesoADecimal*, que recibe (como vector de bits) una representación en Exceso a M de un entero, y devuelve el decimal correspondiente. Ejemplos:

```
>> excesoADecimal([1,0,0,1])    >> excesoADecimal([0,0,0,0,0,1])
ans = 1                         ans = -31
```

Nota: El Exceso a M se calcula con $M = 2^{n-1}$, donde n es la cantidad de bits de la representación.

c) la función ***iterativa*** *flotanteADecimal* que recibe (como vector de bits) una representación en Punto Flotante de un número real, y un natural *exp* indicando la cantidad de bits del exponente, y devuelve el decimal correspondiente. La representación en Punto Flotante tiene las siguientes características:

- Solo considera valores "normalizados".
- El exponente se encuentra en Exceso a M.
- El cero se representa con el exponente y la mantisa en cero.
- No hay representación para infinito ni para "not a number".
- El orden en el vector de bits es signo, exponente y mantisa.

Ejemplos:

```
>> flotanteADecimal([1,0,0,1,0,0,1,0],4)      >> flotanteADecimal([0,0,1,0,0,1],2)
ans = -0.01953125 (igual a -10 x 2^-9)         ans = 0.5625 (igual a 9 x 2^-4)
```

Nota: El Exceso a M se calcula con $M = 2^{n-1}$, donde n es la cantidad de bits de la representación.

COMPUTACIÓN 1
Instituto de Computación

Problema 3 (6, 7 y 7 puntos, respectivamente)

Un observatorio astronómico quiere utilizar Matlab para trabajar con fotografías a cielo abierto, tomadas por el telescopio. Las fotografías se almacenan como matrices de tamaño 1024x1024, donde cada coeficiente de la matriz se corresponde con un pixel de la imagen capturada. Debido a que la cámara solo captura la intensidad, o brillo, de la luz (no los colores), los coeficientes son valores enteros que varían de 0 a 255. El 0 representa el valor de menor intensidad (cielo oscuro) y el 255 el de mayor intensidad (cuerpo celeste muy luminoso).

El observatorio quiere hacer un conteo y una clasificación de los cuerpos celestes que aparecen en las imágenes, partiendo de la suposición de que estos jamás superan el tamaño de 1x1 píxeles. Teniendo en cuenta que los cuerpos se clasifican según la escala de intensidad definida en la siguiente tabla, resuelva los problemas que se plantean a continuación:

Escala	Intensidad	Tipo de cuerpo
1	200-255	Muy brillante
2	150-199	Brillante
3	100-149	Normal
4	50-99	Poco brillante
5	0-49	Cielo oscuro

a) Para realizar el conteo, implemente la función ***recursiva*** *contarCuerposCelestes* que cuente los cuerpos celestes que se encuentren dentro del rango 1-4 de la escala (es decir, que su intensidad sea mayor que 49 y menor o igual que 255).

b) Debido a que la mayoría de los píxeles capturados son de cielo oscuro (escala 5) se quiere comprimir la imagen, convirtiéndola de matriz completa a una matriz dispersa que descarte estos valores de intensidad (es decir, los tome como si fueran iguales a cero). Implemente la función ***iterativa*** *comprimirImagen* que reciba una imagen y devuelva una matriz dispersa que descarte los cuerpos de escala 5.

c) Para clasificar los cuerpos celestes, a partir de una imagen en su forma comprimida implemente la función ***recursiva*** *clasificarCuerposCelestes* que devuelva un vector de cuatro elementos, donde el elemento de la posición *i* cuente la cantidad de cuerpos celestes que corresponden al valor *i* de la escala.

Notas: - La matriz dispersa de la parte b) debe representarse en formato elemental.
- Para la parte c) interesa clasificar solamente los cuerpos celestes correspondientes a las escalas 1 a 4.