

Examen - Febrero de 2010

- Duración del examen: **3 horas**
- No se podrá utilizar **ningún tipo de material** (apuntes, libro, calculadora, etc.)
- Si tiene teléfono **celular, apáguelo**
- Sólo se contestarán preguntas sobre interpretación de la letra hasta 20 minutos antes de la finalización del mismo
- Escriba las hojas de **un solo lado**
- Las partes no legibles del examen se considerarán no escritas
- **IMPORTANTE:** En la primer hoja a entregar escriba con letra clara, en el ángulo superior derecho, salón en el cual desarrolló la prueba, su nombre, número de cédula de identidad y cantidad de hojas entregadas -en ese orden-; en las demás hojas es suficiente escribir nombre, número de cédula y número de página

PROBLEMA 1 (4, 4, 4, 4, 4)

- A) Calcule la expresión decimal del siguiente número binario puro: 101110. Justifique.
- B) Calcule la expresión octal de 1101011010101_2 (en base 2). Justifique.
- C) Represente en complemento a 2 con 5 bits, del número -11. Justifique.
- D) Determine la expresión decimal que representa la tira 0 10000001 100000000000000000000000 en punto flotante simple precisión. Justifique.
- E) Determine la representación en el sistema de punto flotante de -11. Justifique.
- F) Nombre 3 tipos de memoria, especificando en cada caso si es volátil o no y si es solo de lectura o lectura y escritura

PROBLEMA 2 (4, 6, 12)

Se representa un laberinto mediante una matriz de ceros y unos. Si en una celda hay un 1, quiere decir que en ese lugar hay pared; mientras que si hay un cero, quiere decir que por esa celda se puede transitar. En estos laberintos uno solo se puede mover horizontal o verticalmente. Cada movimiento se representa con un número según la orientación que le corresponde:

Arriba: 8
 Abajo: 2
 Izquierda: 4
 Derecha: 6

Un lugar de salida del laberinto es cualquier celda con valor cero que se encuentre en el borde del laberinto. Es decir el la primera o última fila o columna de la matriz.

Ejemplo:

	1	2	3	4	5	6	7	8	9	10
1	[1	1	1	1	1	1	1	1	1	1]
2	[1	1	1	1	1	1	1	1	1	1]
3	[1	1	1	1	0	0	0	0	0	1]
4	[1	0	1	1	0	1	1	1	0	1]
5	[1	0	0	0	0	1	1	1	0	1]
6	[1	1	1	1	0	1	1	1	0	1]
7	[1	1	1	1	0	0	1	1	0	1]
8	[1	1	1	1	1	0	0	1	0	1]
9	[1	1	1	1	1	1	1	1	0	1]

En este laberinto, si nos encontramos en la celda de coordenadas (5, 5) el camino de salida es el siguiente:

[8, 8, 6, 6, 6, 6, 2, 2, 2, 2, 2, 2]. El único lugar de salida de este laberinto es mediante la celda (9, 9).

Se pide:

- A) Implementar una función en *Matlab* que dado un laberinto y una posición devuelva 1 si la posición es una salida.
- B) Implementar una función en *Matlab* que dado un laberinto, una posición dada por un vector de dos columnas y un movimiento representado por un número, devuelva dos valores: un valor de éxito (1 o 0) si se puede realizar dicho movimiento (si no es pared la posición original, si no es pared el destino, si esta dentro del tablero, etc.) y el destino del movimiento.
- C) Implementar una función **recursiva** en *Matlab* que dados un laberinto, una posición inicial y un vector de movimientos, devuelva dos valores: un valor de éxito (1 si el camino se pudo recorrer completo o 0 en caso contrario) y el destino final de aplicar todos los movimientos (en caso de poder efectuarlos). *NOTA:* Puede utilizar la parte B.

PROBLEMA 3 (3, 9, 12)

Una matriz de permutación P es una matriz cuadrada que cumple con los siguientes requisitos:

- En cada fila hay un solo valor en 1, y los demás son 0
- No hay ninguna columna que tenga dos valores 1

Si la matriz P es de tamaño $n \times n$ y tenemos una matriz A de tamaño $n \times m$, se cumple que el producto $P \cdot A$ tiene como resultado una matriz que tiene los mismos elementos que A pero realizando una permutación de sus filas. Por ejemplo:

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, A = \begin{bmatrix} 2 & 6 & -4 & 2 \\ 3 & -1 & 2 & 6 \\ -5 & 3 & 3 & 1 \end{bmatrix}, A \times P = \begin{bmatrix} -5 & 3 & 3 & 1 \\ 2 & 6 & -4 & 2 \\ 3 & -1 & 2 & 6 \end{bmatrix}$$

El producto $P \cdot A$ se puede realizar sin calcular todas las operaciones del producto de matrices común. Notar que si la matriz P tiene un elemento 1 en la posición (i, j) , entonces la fila j de la matriz A se corresponderá con la fila i de la matriz $P \cdot A$.

En el ejemplo, el resultado está formado por: la tercera fila de A , luego la primera fila de A , y luego la segunda fila de A .

Como las matrices de permutación tienen (por definición) una gran cantidad de elementos con valor 0, resultan ideales para almacenarse utilizando la representación dispersa.

Se pide:

- A) Muestre como es la matriz P del ejemplo en formato disperso elemental
- B) Implemente una función **iterativa** `permutadc` en *Matlab* que tome como entrada una matriz de permutación P representada como matriz dispersa en formato elemental de tres vectores y una matriz completa A , y devuelva una matriz completa que sea equivalente al resultado del producto $P \cdot A$.
- C) Implemente una función **iterativa** `permutadd` en *Matlab* que tome como entrada una matriz de permutación P representada como matriz dispersa y una matriz A representada como matriz dispersa, y devuelva una matriz dispersa que sea equivalente al resultado del producto $P \cdot A$. Las matrices P , A y el resultado deben estar representadas en el formato elemental de tres vectores.

NOTA: Para este ejercicio no se permite utilizar el producto de matrices de Matlab (operador *), ni ninguna otra función que resuelva trivialmente el problema. Recuerde que el formato disperso elemental no especifica ningún orden particular, por lo cual en la parte C del ejercicio no necesita ordenar el resultado.

PROBLEMA 4 (7, 9, 14)

Considere la siguiente secuencia s : Partiendo de un entero s_0 , sea s_1 la suma de los cuadrados de los dígitos de s_0 . De forma similar, sea s_2 la suma de los cuadrados de los dígitos de s_1 , y así sucesivamente. Por ejemplo, comenzando con $s_0 = 7$ se obtiene la secuencia:

$$s_0 = 7$$

$$s_1 = 7^2 = 49$$

$$s_2 = 4^2 + 9^2 = 97$$

$$s_3 = 9^2 + 7^2 = 130$$

$$s_4 = 1^2 + 3^2 + 0^2 = 10$$

$$s_5 = 1^2 + 0^2 = 1$$

Es claro que si para algún i se cumple $s_i = 1$, entonces para todo j mayor o igual que i también se cumplirá $s_j = 1$. Si existe para esta secuencia un i tal que $s_i = 1$, se dice que el número s_0 que inició la secuencia es un *número feliz*.

Se pide:

- A) Implementar en *Matlab* una función **iterativa** `separardigitos` que, dado un entero no negativo, retorne un vector donde cada elemento es un dígito del entero.

Ejemplo:

```
>> separardigitos(49)
ans = [4, 9]
```

- B) Implementar en *Matlab* una función **iterativa** `secuencia` que reciba como parámetro un entero no negativo y devuelva el siguiente valor de la secuencia s definida anteriormente.

Ejemplo:

```
>> secuencia(97)
ans = 130
```

- C) Implementar en *Matlab* una función **recursiva** `esfeliz` que tome como entradas un entero no negativo n y un valor `tope`. Esta función debe calcular como máximo `tope` valores de la secuencia s comenzando con el número n . En caso de encontrar un valor de la secuencia que valga 1, no se debe seguir calculando los siguientes valores. La función `esfeliz` debe retornar una salida compuesta por dos resultados:

- Si la función calculó `tope` valores y no llegó a ningún 1, el primer parámetro de salida debe ser 0. En caso contrario se demostró que el número ingresado es feliz, por lo tanto el primer parámetro de salida debe valer 1.
- En cualquier caso, en el segundo parámetro de salida se debe retornar un vector comenzando con el valor n y conteniendo todos los valores de la secuencia calculada.

Ejemplo:

```
>> [f,s] = esfeliz(7,100)
f = 1
s = [7, 49, 97, 130, 10, 1]
```