

1^{er} Parcial - 04 de Octubre de 2014

- 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 30 minutos antes de la finalización del mismo.
- Escriba las hojas de un solo 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, el salón en el cual desarrolló la prueba, su nombre, número de cédula de identidad y cantidad de hojas -en ese orden-; en las demás hojas es suficiente que ponga nombre, número de cédula y número de página.

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

- `length()`
- `size()`
- `mod()`

Problema 1	8 (4,4) pts	
-------------------	-------------	--

a) Suponga que no se dispone de `&` ni de `~` en Matlab. Escriba las funciones **`and(x,y)`** y **`not(x)`**, asumiendo que **`x`** e **`y`** sólo contienen valores booleanos (**`0`** ó **`1`**), utilizando instrucciones de selección (**`if`**, **`if-else`** o **`elseif`**).

b) Dado el siguiente código:

```

if ( a > b )
    if ( c <= d )
        x = 1;
    else
        x = 2;
    end

    if ( d == a )
        x = 2;
    end
else
    if ( b > d )
        x = 3;
    end
    x = 4;
end

```

- Escriba una expresión lógica (utilizando las variables **`a`**, **`b`**, **`c`** y **`d`**) que represente los casos en los que la variable **`x`** termina con el valor **`1`**.
- Escriba una expresión lógica (utilizando las variables **`a`**, **`b`**, **`c`** y **`d`**) que represente los casos en los que la variable **`x`** termina con el valor **`2`**.
- Escriba una expresión lógica (utilizando las variables **`a`**, **`b`**, **`c`** y **`d`**) que represente los casos en los que la variable **`x`** termina con el valor **`4`**.
- Escriba una expresión lógica (utilizando las variables **`a`**, **`b`**, **`c`** y **`d`**) que represente los casos en los que la variable **`x`** termina con el valor **`3`**.

Problema 2	6 (2,2,2) pts	
-------------------	---------------	--

- a) Determine la representación en base 2 del número que en base 10 se representa como 17.
- b) Determine la representación en base 10 del número que en base 8 se representa como 65.
- c) Determine la representación en binario del número que en base 16 se representa como 101.

Problema 3	12 (5,5,2) pts	
-------------------	----------------	--

- a) Implemente una función en Matlab que dado un vector de números naturales, devuelva la suma de todos los elementos del vector, excepto aquellos múltiplos de 7.
- b) Implemente una función en Matlab que dado un vector de números naturales, calcula la suma de los elementos del vector, excepto aquellos múltiplos de 7, hasta que la suma supere 60.
- c) Implemente una función en Matlab que dado un vector de números naturales y dos números naturales (**a** y **b**), calcula la suma de los elementos del vector, excepto aquellos múltiplos de **a**, hasta que la suma supere el valor de **b**.

Problema 4	14 (5,2,7) pts	
-------------------	----------------	--

- a) Implemente una función en Matlab que dado un vector \mathbf{v} (no vacío), devuelva el mínimo elemento del vector y en qué posición del vector está. Si el mínimo elemento se encuentra repetido en el vector, se deberá devolver la posición correspondiente a la primera ocurrencia de izquierda a derecha.
- b) Implemente una función en Matlab que dado un vector \mathbf{v} , devuelva un vector con los mismos elementos de \mathbf{v} pero con el siguiente intercambio, en la primera posición (posición más a la izquierda) estará el valor mínimo del vector original, y en la posición que estaba el mínimo estará el valor que estaba en la primera posición del vector original.
- c) Implemente una función en Matlab que dado un vector \mathbf{v} de largo \mathbf{n} , devuelva un vector con los mismos elementos de \mathbf{v} pero ordenados de menor a mayor, de acuerdo al mecanismo que se describe a continuación. Primero se busca el mínimo elemento del vector y se intercambia con el primer elemento de ese vector. Luego se busca el mínimo elemento en el resto del vector (a partir de la segunda posición) y se intercambia con el segundo elemento del vector. En el caso general, en el paso \mathbf{i} se busca el mínimo elemento entre una posición \mathbf{i} y el final del vector, y se intercambia el mínimo encontrado con el elemento de la posición \mathbf{i} . Después de $\mathbf{n}-1$ pasos, el vector resultante está ordenado.

Nota: En la parte b) se puede usar la parte a). En la parte c) se puede usar la parte b). No se pueden usar las funciones `min()` ni `sort()` de Matlab.
