

1^{er} Parcial - Setiembre de 2010 - 2^a 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 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, salón en el cual desarrolló la prueba, 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 (4, 4, 5) pts	
-------------------	------------------	--

Se desean realizar diferentes cálculos con un vector de números reales. Para ello, se pide:

- a) Escribir en *Matlab* una función que reciba un vector v de números reales como parámetro de entrada y devuelva como parámetro de salida el resultado de calcular el promedio de los elementos del vector cuyos valores sean positivos.
- b) Modificar la función de la parte a) para que devuelva dos parámetros de salida con los resultados de calcular, respectivamente: 1) el promedio de los elementos con valor positivo en posiciones pares, 2) el promedio de los elementos con valor positivo en posiciones impares. La función resultante **NO** puede recorrer dos veces el vector para calcular el resultado.
- c) Escribir una función en *Matlab* que cuente y sume los primeros 10 números pares de un vector v y devuelva estos resultados en dos parámetros de salida.

Ejemplos:

```
>> a = [0 1 0 -1 -2 -2 1 -1];
>> res = parteA(a)

res =
    0.5000

>> [res1,res2] = parteB(a)

res1 =
     1
res2 =
    0.3333

>> c = [2 0 3 1 7 9 8 4 5]
>> [res1,res2] = parteC(a)

res1 =
     4
res2 =
    14
```

- Notas:**
- Al valor 0 (cero) se lo debe considerar tanto POSITIVO como NEGATIVO a la hora de calcular los promedios.
 - Si al calcular un promedio (ej.: promedio de valores positivos) el vector no contiene ningún valor que cumpla con la condición (ej.: ser positivo), se debe retornar 0 (cero).
 - Se puede utilizar la función de *Matlab* módulo, `mod(divisor,dividendo)`, que retorna el resto de la división entera divisor/dividendo para determinar si un número es par o impar.
 - Para la parte c) el vector puede contener menos de 10 pares. En ese caso, la función debe contar y sumar la cantidad de pares existente.

Problema 2	8 pts	
-------------------	-------	--

Se desea verificar, mediante una función implementada en *Matlab*, si el registro contable de una cuenta monetaria es correcto. El registro contable es representado por tres vectores de idéntico tamaño: *Debe*, *Haber* y *Saldo*. Además, se conoce el saldo inicial de la cuenta bancaria antes de que se efectuaran los movimientos especificados en el registro contable. Los valores en el vector *Debe* representan retiros de la cuenta. Los valores en el vector *Haber* indican depósitos en la cuenta. Mientras que el *i*-ésimo valor del vector *Saldo* representa el saldo de la cuenta luego de haber realizado todos los movimientos indicados en los vectores *Debe* y *Haber* hasta la posición *i*, inclusive.

Ejemplo: Este ejemplo presenta un registro contable correcto (todos los saldos son correctos).

Saldo inicial: 1000

Posición	Debe	Haber	Saldo
1	100	0	900
2	0	200	1100
3	200	0	900
4	100	400	1200
5	175	175	1200

El objetivo de la función es verificar que el saldo especificado es el correcto luego de haber efectuado cada uno de los movimientos indicados en los vectores *Debe* y *Haber*.

Se pide: Implemente la función *verificarRegistro*, que recibe como parámetros el saldo inicial y los tres vectores *Debe*, *Haber* y *Saldo*; y verifica que el contenido del vector *Saldo* es correcto. La función debe devolver un número indicando la primera posición en la que se detectó que el saldo es incorrecto. Si todos los saldos son correctos, la función debe devolver cero.

Problema 3	9 pts	
-------------------	-------	--

Dadas dos matrices $A_{m \times n}$ y $B_{n \times p}$, y dos números *min* y *max*, se desea implementar la función ***mulMatRec*** que realiza la multiplicación de las matrices *A* y *B* “recortando” el valor de cada uno de los coeficientes, de forma que no sean mayores que el número *max* y que no sean menores que el número *min*. En otras palabras, para cada coeficiente $M(i,j)$ de la matriz $M=AxB$: si $M(i,j)$ resulta mayor que *max* entonces se le asignará *max* al coeficiente (i,j) de la matriz resultante; si $M(i,j)$ es menor que *min*, se la asignará *min* al coeficiente (i,j) de la matriz resultante; en otro caso, el coeficiente (i,j) tomará el valor $M(i,j)$.

Recuerde que el producto entre dos matrices se calcula como $M(i,j) = \sum_{k=1}^{k=n} A(i,k) \times B(k,j)$.

La función ***mulMatRec*** debe retornar una matriz *R*, resultado de la multiplicación “recortada”.

Se pide implementar en *Matlab* la función ***mulMatRec*** antes descrita.

Ejemplo:

$$\text{mulMatRec} \left(\left(\begin{bmatrix} 1 & 5 & -3 \\ -1 & 2 & 3 \\ -5 & -2 & 1 \end{bmatrix}, \begin{bmatrix} -2 & 3 & 5 \\ 1 & -2 & 3 \\ 5 & -2 & 1 \end{bmatrix}, -15, 15 \right) = \begin{bmatrix} -12 & -1 & 15 \\ 15 & -13 & 4 \\ 13 & -13 & -15 \end{bmatrix} \right.$$

Nota: No se permite el uso de la multiplicación de matrices provista por *Matlab*.

Asuma que las dimensiones de las matrices *A* y *B* siempre son adecuadas para la multiplicación: $A_{m \times n}$ y $B_{n \times p}$.