



2^{do} Parcial - Diciembre de 2008 - 2^a parte

- Duración del parcial: 3 Hs.
- 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	15 (8,7) ptos	
------------	---------------	--

a) Implementar en *Matlab* una **función recursiva** llamada “**separarParesImpares**”. Dicha función recibe como único parámetro de entrada un vector de enteros; y devuelve como parámetros de salida dos vectores agrupando los números pares en el primer parámetro de salida, y los números impares en el segundo parámetro de salida. Se tiene que respetar el orden relativo entre los elementos de los vectores.

Ejemplo:

```
>>[a b] = separarParesImpares([1,8,6,2,1,7,9,3,5,1,9,2,7])
```

```
a: [8,6,2,2]
```

```
b: [1,1,7,9,3,5,1,9,7]
```

b) Implementar en *Matlab* la **función recursiva** llamada “**juntarParesImpares**”. La misma recibe exactamente dos parámetros de entrada: el primer parámetro es un vector de enteros pares, y el segundo parámetro es un vector de enteros impares. La función deberá intercalar los elementos de ambos vectores para generar un vector en donde su primer elemento es par, el siguiente es impar, el siguiente es par... y así análogamente hasta que se terminen los elementos de una de los dos vectores de entrada. Desde ese punto en adelante todos los elementos del vector resultado van a ser solo pares o solo impares.

Ejemplo:

```
>>resultado = juntarParesImpares([8,6,2,2],[1,1,7,9,3,5,1,9,7])
```

```
resultado: [8,1,6,1,2,7,2,9,3,5,1,9,7]
```

Nota: Para la resolución de este ejercicio **NO** se permite el uso de ninguna función que por su naturaleza, resuelva trivialmente el problema.

Problema 2 17 (9, 8) ptos

a) Considere una función $f: \mathbf{N} \rightarrow \mathbf{R}$. Podemos representar los n primeros valores de dicha función mediante un vector en *Matlab* [f_1 f_2 ... f_n].

Se denomina *derivada discreta de f en i* (se nota como Df_i) al valor $f_{i+1} - f_i$

Escriba una función recursiva “**derivadadiscrec**” en *Matlab* que tome como entrada un vector que contenga los n primeros valores de f y devuelva un vector que represente los $n-1$ primeros valores de la *derivada discreta* de f .

b) Dado un vector Df que representa la *derivada discreta* de f y una constante C , es posible hallar la *integral discreta de Df en i* (se nota como S_i) mediante la recurrencia:

$$S_1 = C$$

$$S_{i+1} = S_i + Df_i, \text{ para } i \geq 1$$

Escriba una función recursiva “**integraldiscrec**” en *Matlab* que tome como entrada un vector Df y una constante C y devuelva un vector con los valores [S_1 S_2 ... S_n].

Notas:

- Observar que realizando una elección apropiada de la constante C (específicamente tomando $C=f_1$), la *integral discreta* de todos los elementos del vector Df dará como resultado el vector original con los n primeros elementos de f .
- En este ejercicio **NO** se permite utilizar la función `sum` ni ninguna otra función de *Matlab* que, por su naturaleza, resuelva trivialmente el problema.

Problema 3 18 (4, 6, 8) ptos

Para comprimir una matriz completa que contiene datos que son generalmente iguales (con poca o casi nula variación entre ellos) vamos a utilizar la siguiente metodología:

1. Determinar la frecuencia de ocurrencia de cada uno de los datos que aparece en la matriz.
2. Obtener el dato más frecuente.
3. Restar a la matriz original el dato hallado en el punto anterior.
4. Convertir la nueva matriz (resta) completa en la representación a 3 vectores para matices dispersas vista en el curso.

Se pide por lo tanto que realice las siguientes funciones en *Matlab*:

- a) Escribir una función **iterativa** llamada “**DatoFrec**” que dada una matriz completa devuelva una matriz de 2 columnas y N filas. Esta función debe calcular la frecuencia de aparición de los diferentes elementos (datos) que se encuentran en la matriz. En la primer columna debe aparecer el dato y en la segunda su frecuencia (o número de apariciones). En caso de invocarla con una matriz nula devolver vector vacío.
- b) Escribir una función **iterativa** llamada “**MasFrec**” que dada la matriz calculada en la parte a), devuelva la fila correspondiente al elemento más frecuente. Si existe más de uno con igual frecuencia devolver cualquiera de ellos.
- c) Escribir una función **iterativa** llamada “**Descomponer**” que dada una matriz completa realice la descomposición en los 3 vectores vistos en el curso para representarla como si fuese una matriz dispersa. En caso de invocarla con una matriz nula devolver los 3 vectores vacíos.

Nota: No se puede usar `sort`, `find`, `sparse`, ni ninguna otra función que resuelva trivialmente el problema.