

Examen - 19 de febrero de 2021
Parte 3: Recursión – 36 puntos

- Duración de esta parte de la prueba: 60 minutos.
- Duración total incluyendo descansos: 200 min (45' – 10' – 75' – 10' - 60') = 3h:20min
- Puntaje total de la prueba: 100 puntos.
- 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.

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

- length() y size()
- mod() y rem()
- floor(), ceil() y round()
- zeros() y ones()

Notas: - En todos los ejercicios se deben usar las estructuras de control adecuadas para cada caso.
- No se deben realizar más invocaciones recursivas que las necesarias para resolver el problema

Problema 7 | 12 ptos

Los coeficientes binomiales (o combinaciones) corresponden al número de formas en que se puede extraer subconjuntos a partir de un conjunto dado. Es decir $C(n, k)$ denota el número de formas de elegir k elementos a partir de un conjunto de n elementos ($k \leq n$). Los coeficientes binomiales se pueden definir de forma recursiva de la siguiente forma para $n \geq 0$ y $k \geq 0$ (con n y k enteros):

- 1) $C(n, k) = 1$ para $k = 0$.
- 2) $C(n, k) = C(n-1, k-1) + C(n-1, k)$ para cualquier par de enteros n y k que cumplan $1 \leq k \leq n-1$.
- 3) $C(n, k) = 1$ para $k = n$.

Implementar en *Octave* una función **recursiva binomial** que dado un par de enteros n y k calcule el coeficiente binomial correspondiente. La función debe retornar 0 si los parámetros no son válidos para la función.

En los siguientes ejemplos el primer parámetro es n y el segundo parámetro es k .

binomial(-2,3) devuelve 0	binomial(2,0) devuelve 1
binomial(2,-1) devuelve 0	binomial(2,1) devuelve 2
binomial(0,1) devuelve 0	binomial(2,2) devuelve 1
binomial(2,3) devuelve 0	binomial(3,0) devuelve 1
binomial(0,0) devuelve 1	binomial(3,1) devuelve 3
binomial(1,0) devuelve 1	binomial(3,2) devuelve 3
binomial(1,1) devuelve 1	binomial(3,3) devuelve 1

Problema 8 | 12 ptos

Implementar en *Octave* una función **recursiva comienzaCon** que, dado dos vectores $v1$ y $v2$, devuelva 1 si el vector $v1$ comienza con el vector $v2$, y 0 en caso contrario. Notar que ambos vectores pueden tener cualquier tamaño. El vector vacío es comienzo de cualquier vector.

Ejemplos:

```

comienzaCon([], []) devuelve 1
comienzaCon([1,7,2,4], []) devuelve 1
comienzaCon([1,7,2,4], [1]) devuelve 1
comienzaCon([1,7,2,4], [7]) devuelve 0
comienzaCon([1,7,2,4], [1,7]) devuelve 1
comienzaCon([1,7,2,4], [1,2]) devuelve 0
comienzaCon([1,7,2,4], [1,7,4]) devuelve 0

```

Problema 9	12 pts
-------------------	--------

Implementar en *Octave* la función **recursiva** *multiplicacionElementoaElementoFiltrada*, que dadas dos matrices **A** y **B** completas y una matriz **S** dispersa en **formato elemental**, devuelva una matriz dispersa en **formato elemental D** que contenga el producto "elemento a elemento" de **A** y **B** en las posiciones de **S** que son distintas de 0, siempre y cuando dicho producto no dé 0. Es decir $D(i,j)=A(i,j)*B(i,j)$ si y solo si existe un elemento distinto de cero en la matriz dispersa **S** en la posición (i,j) (fila i y columna j) y el producto $A(i,j)*B(i,j)$ es distinto de 0. **A** y **B** tienen exactamente las mismas dimensiones. Los valores de las filas y las columnas de los elementos no ceros de **S** están dentro de las dimensiones de **A** y **B**.