Examen - 28 de diciembre de 2020
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	

Sea un vector ***v*** que contiene números y un vector ***mascara*** que contiene unos y ceros y tiene exactamente el mismo largo que el vector ***v***. Implementar en *Octave* la función ***recursiva*** *recortarMascara* que, dados ***v*** y ***mascara***, devuelva el vector ***v*** pero habiendo eliminado los elementos de las posiciones en las que el valor de ***mascara*** sea 1, es decir se elimina el elemento ***v***(i) siempre y cuando **mascara**(i) sea 1. Los vectores pueden ser vacíos.

En los siguientes ejemplos el primer vector es ***v*** y el segundo vector es ***mascara***.

```
RecortarMascara([],[]) devuelve []
recortarMascara([1,2,3,4],[1,0,0,1]) devuelve [2,3]
recortarMascara([10,-2,5,3,-1],[0,1,0,1,1]) devuelve [10,5]
```

Problema 8	12 ptos	

Sea un vector ***v*** que contiene números y un vector ***posiciones*** que contiene números enteros mayores o iguales a 1 y menores o iguales al largo del vector ***v***. Implementar en *Octave* la función ***recursiva*** *sumarPosiciones* que, dados ***v*** y ***posiciones***, devuelva el resultado de sumar los elementos de las posiciones indicadas en el vector ***posiciones***. Note que los vectores no tienen porque tener el mismo largo y que pueden haber elementos repetidos en ***posiciones***.

En los siguientes ejemplos el primer vector es ***v*** y el segundo vector es ***posiciones***.

```
SumarPosiciones([1,5,3],[]) devuelve 0
sumarPosiciones([4,3,5,2],[1]) devuelve 4
sumarPosiciones([11,2,-3],[3,1,2,1]) devuelve 21
sumarPosiciones([1,-4,2,6,-1,7],[3,5,3,1,5]) devuelve 3
```

Problema 9	12 ptos	

Implementar en *Octave* la función ***recursiva*** *triptico* que, dado un número ***N*** entero positivo mayor o igual a 1, devuelva tres vectores *ordenados* conteniendo los números entre 1 y ***N*** inclusive, clasificados de acuerdo al resto de la división entre 3. Es decir, uno de los vectores contiene todos los números entre 1 y ***N*** que son múltiplos de 3, otro vector contiene todos los números entre 1 y ***N*** cuyo resto es 1 al dividirlos entre 3, y otro con los que el resto es 2.

```
triptico(1) devuelve [], [1] y []
triptico(3) devuelve [3], [1] y [2]
triptico(10) devuelve [3,6,9], [1,4,7,10] y [2,5,8]
triptico(20) devuelve [3,6,9,12,15,18], [1,4,7,10,13,16,19] y
                      [2,5,8,11,14,17,20]
```