

Parte 2:

- Duración de esta parte: **75 min.**
- Duración total incluyendo descanso: **160 min** (75' - 10' - 75') = 2h:40min
- Puntaje total de la prueba: **100 puntos.**
- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc).
- El celular solo utilícelo si lo precisa para conectarse a Zoom. De lo contrario apáguelo.
- **Sólo** se contestarán preguntas sobre interpretación de la letra hasta 20 minutos antes de la finalización de cada parte.

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()`
- `abs()`
- `zeros()` y `ones()`

Notas:

- **No realizar más iteraciones ni invocaciones recursivas que las necesarias para resolver el problema**
 - **En todos los ejercicios se deben usar las estructuras de control adecuadas para cada caso.**
- Por ejemplo: se controlará el uso correcto de for y while.**

Problema 4	16 pts	
-------------------	--------	--

Una celda (i,j) de una matriz se dice que tiene *posición par* si $i+j$ es par.

Escriba en Octave la función **iterativa** `restaParIt` que, **dada una matriz dispersa en formato elemental, devuelva otra matriz dispersa** donde a cada posición par representada en la primera matriz se le haya restado 1 al valor de la celda.

Problema 5	16 pts	
-------------------	--------	--

Dada una matriz **M** **compuesta por ceros y unos**, decimos que hay *parejas de unos consecutivos* cuando hay al menos dos celdas de la misma fila i que cumplen $M(i, j) = M(i, j+1) = 1$.

Escriba en Octave la función **iterativa** `HP=HayParejasIt(M)`, que dada una matriz M de dimensiones $m \times m$ **densa y cuadrada**, devuelve un vector `HP` de largo m , tal que $HP(k) = 1$ si en la fila k de M existe al menos un par de unos consecutivos. En caso contrario $HP(k) = 0$.

Ejemplos:

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow HP = [1 \ 0 \ 0] ; \quad M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \Rightarrow HP = [1 \ 0 \ 1 \ 1] ;$$

$$M = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow HP = [0 \ 0 \ 0] ;$$

$$M = [1] \Rightarrow HP = [0] ; \quad M = [0] \Rightarrow HP = [0] ;$$

Problema 6	18 pts	
-------------------	--------	--

Escriba en Octave la función **recursiva** `res=SumPolisRec(p1,p2)`, que dado dos polinomios $p_1(x)$ y $p_2(x)$ representados como vectores `p1` y `p2`, devuelva la suma de ambos. Para simplificar el código, asuma que el grado de $p_1(x)$ es mayor al grado de $p_2(x)$.

```
Ej: res=SumPolisRec([1 0 1],[1 2])    retorna [1 1 3]
     res=SumPolisRec([1 0 1],[-1])    retorna [1 0 0]
     res=SumPolisRec([1 0 1],[0])     retorna [1 0 1]
     res=SumPolisRec([1 0 1],[ ])    retorna [1 0 1]
     res=SumPolisRec(2,[ ])          retorna [2]
```