



(0612) PROGRAMACIÓN II  
(1110) PROGRAMACIÓN  
FINAL  
MESA A  
14 hs  
22/10/ 2021

Apellido y Nombre:

DNI:

PARCIAL:

Calificación :

## Ejercicio C

Dado un archivo de texto "mat1.txt" que contiene una matriz rara.

Una **matriz rara** o **matriz dispersa** o **matriz hueca** es una matriz de gran tamaño en la que la mayor parte de sus elementos son cero.

### Matriz Dimensiones:

- $0 < \text{filas} < 1.000$
- $0 < \text{columnas} < 1.000$

### Archivo:

- La primera línea del archivo indica las dimensiones de la matriz de la forma:  
[filas][columnas]  
Los corchetes aparecen de forma literal en el archivo de texto.
- El resto de las líneas: indica los valores no nulos de la matriz, el formato es el siguiente;  
[fila][columna] valor no nulo  
Estas líneas no tienen ningún orden.

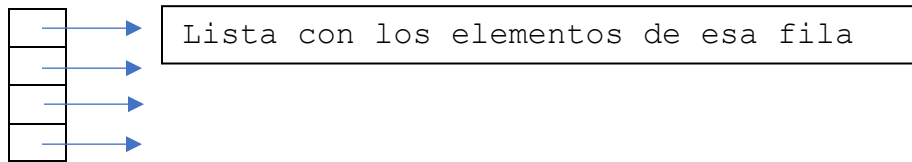
**Objetivo propuesto:** desarrollar una aplicación, main y la/s función/es necesarias, determinar el mayor elemento de la diagonal principal de una matriz rara. Su solución debe utilizar las siguientes funciones, las cuales debe desarrollar correctamente:

**main:** debe optimizar el proceso y el uso de las funciones a desarrollar para lograr el objetivo propuesto.

**verificar\_mat:** esta función verifica si es posible alcanzar el objetivo propuesto. Esta función debe invocarse antes de cargar la matriz en memoria.

**cargar\_mat\_rala:** función para cargar una matriz en memoria.

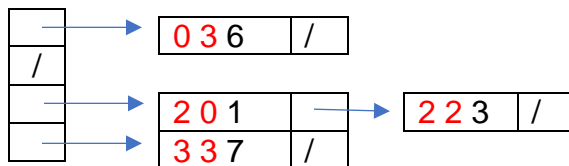
Las matrices solo guardan los valores no nulos. Para hacer esto se propone que guarde las matrices de la siguiente forma:



Ejemplo: Dada la siguiente matriz rala:

	0	1	2	3
0	0	0	0	6
1	0	0	0	0
2	1	0	3	0
3	0	0	0	7

La función debe crear la siguiente estructura:



**maxDP\_mat\_rala:** función que obtiene el máximo elemento de la diagonal principal de una matriz rala. Debe retornar el valor y su ubicación dentro de la matriz.

**imprimir\_mat\_rala:** recibe una matriz rala y la muestra completa (con todos los valores, incluyendo los 0).

## Ejercicio C++

Desarrolle la clase “Medicion” para que el código “main” provisto al pie sea válido. Los objetos de la clase “Medicion” almacenan el valor de una medición en flotante y su unidad de medida como cadena de caracteres. Las mediciones se pueden operar siempre y cuando su unidad de medida sea la misma, caso contrario la operación no es permitida. También es posible, como se indica en el código al pie sumar a un flotante la “Medicion”. En este caso no es importante la unidad de medida ya que existe una sola. Use efectivamente el tiempo y no desarrolle nada que no aplique al código mostrado.

Si el código entregado contiene errores comente la línea indicando cual es el error y luego escriba la línea en forma correcta.

```
Medicion m1mv(100.0, "Mv");//millivolts
Medicion m2mv(20.0, "Mv");
Medicion m4amp(3.0, "Amp");
Medicion m3mv=m1mv-m2mv;
cout<<"Resultado 1: "<<180.0+m3mv<<endl;

try{
    cout<<m3mv-m4amp<<endl;
}catch(...){
    cout<<"No se pueden restar mediciones de distinta unidad de medida"<<endl;
}
```

## EVALUACIÓN

### NOTA GENERAL

- El final se desarrolla con la cámara encendida, durante toda la duración del mismo.
- En caso de ser requerido, debe abrir el micrófono.
- La hora límite de entrega es 17:00 hs.
- Las notas serán enviadas por MIEL dentro de las 72 hs hábiles
- Desarrolle cada ejercicio en un proyecto separado.
- **Incluya en el encabezado de cada archivo, // apellido\_nombre\_DNI**
- Recuerde antes de comprimir, eliminar las carpetas bin y obj de cada proyecto.
- **Entregue ambos proyectos compactados en un zip, “apellido\_nombre\_DNI.zip”.**
- Entregue el parcial usando prácticas de MIEL.
- Enviar a todos los tutores.
- ¡La evaluación es individual!

¡El mayor de los éxitos!