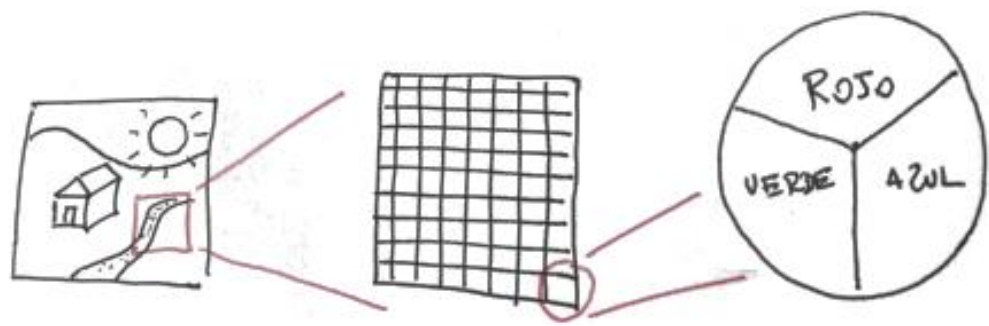


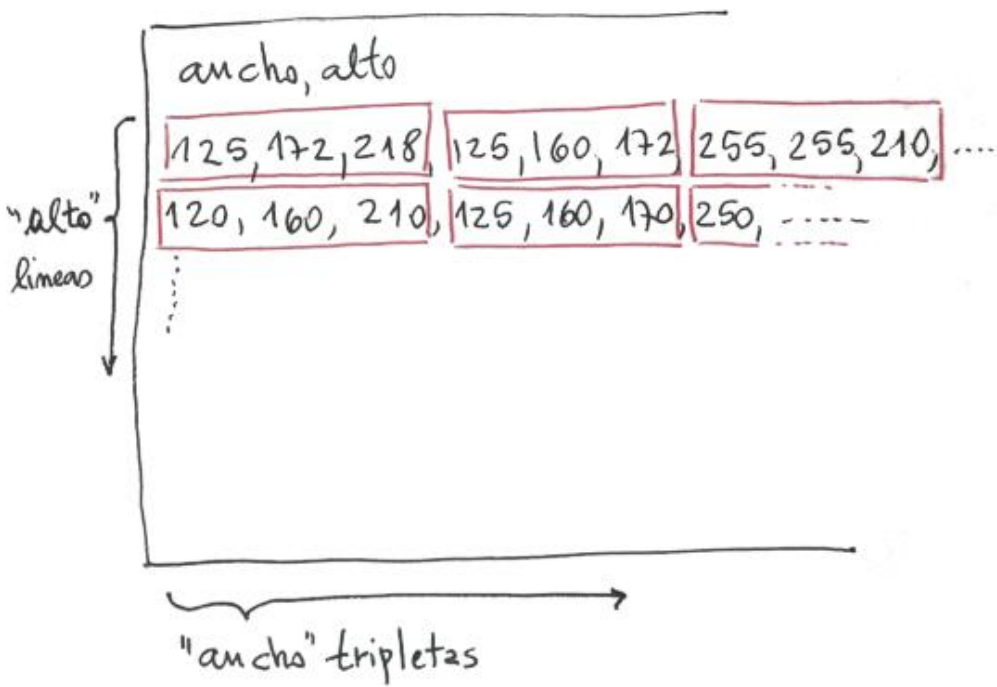
Lea la prueba completamente DOS veces antes de hacer cualquier pregunta

**Problema A. Editor de Imágenes (40%)**

Una imagen se puede pensar como una matriz de elementos, en que cada elemento es un pixel, que a la vez está compuesto de 3 cosas: la cantidad de color rojo, verde y azul. Al mezclar estos colores se obtiene el color de cada pixel de la imagen.



Para este ejercicio, se usarán imágenes almacenada en archivos de texto. Estos archivos tendrán el siguiente formato:



La primera línea contiene el par ancho,alto de la imagen. Estos números corresponden a la cantidad de pixeles de ancho (a lo largo del eje X) y alto (a lo largo del eje Y) de la imagen.

A continuación hay una línea por cada fila de pixeles en la imagen (o sea, hay “alto” líneas). Cada una de estas líneas contiene a su vez “ancho” tripletas de números, con los valores de rojo, verde y azul para ese píxel.

Se pide crear un programa que el ser ejecutado, muestre el siguiente menú de opciones:

# Simulacro – Programación – 202510

**Duración:** 3 horas

16 de junio 2025

Bienvenido al Editor de Imágenes

Seleccione una opción:

C: Cargar imagen

M: Mostrar imagen

B: Aplicar blur

G: Convertir a escala de grises

X: Pixelar

I: Invertir

N: Blanco y negro (normal)

D: Blanco y negro (dither)

S: Salir

Cuando la persona ejecutando el programa escribe alguna de las letras indicadas, se debe llevar a cabo la acción. Si se ingresa una letra que no sea válida, se debe volver a mostrar el menú de opciones.

## C: Cargar imagen

Al seleccionar esta acción, el programa debe primero preguntar el nombre del archivo a cargar, de la siguiente forma:

Ingrese el nombre del archivo: imagen3.txt

Se debe cargar la imagen almacenada en el archivo indicado y guardarla en la memoria. Todas las otras acciones (M, B, G, X, I, N, D) se deben realizar sobre la imagen cargada.

Si se vuelve a seleccionar esta acción, se repite el proceso, se vuelve a preguntar el nombre del archivo y la imagen se lee desde el archivo que se indique.

## B: Aplicar blur

Al seleccionar esta acción, el programa debe cambiar el valor de cada píxel por este nuevo valor:

$$\text{Rojo}(i,j) = \text{Promedio}(\text{Rojo}(i,j)) + \text{Valor Rojo de mis 8 vecinos}$$
$$\text{Verde}(i,j) = \text{Promedio}(\text{Verde}(i,j)) + \text{Valor Verde de mis 8 vecinos}$$
$$\text{Azul}(i,j) = \text{Promedio}(\text{Azul}(i,j)) + \text{Valor Azul de mis 8 vecinos}$$

Esto quiere decir que el valor de cada píxel se debe sobrescribir con el valor promedio de sus 8 vecinos más él mismo. Esto se debe hacer en forma independiente para los tres colores.

NOTA: Para el caso de los píxeles de los bordes, considere que sus vecinos tienen valor cero

## G: Convertir a escala de grises

Al seleccionar esta acción, el programa debe cambiar el valor de cada píxel por este nuevo valor:

$$\text{nuevoColor} = 0.21 \text{ Rojo}(i,j) + 0.71 \text{ Verde}(i,j) + 0.07 \text{ Azul}(i,j)$$

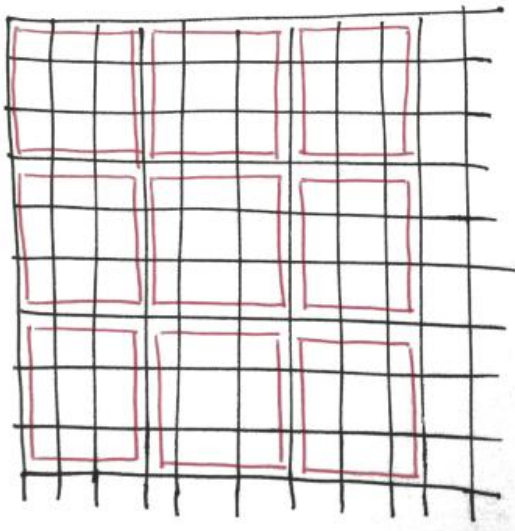
$$\text{Rojo}(i,j) = \text{nuevoColor}$$

$$\text{Verde}(i,j) = \text{nuevoColor}$$

$$\text{Azul}(i,j) = \text{nuevoColor}$$

## X: Pixelar

Al seleccionar esta acción, el programa debe cambiar el valor de cada píxel por el promedio de sus vecinos, tal como se muestra en la siguiente figura:



Dentro de cada subdivisión, se debe calcular el promedio, y todos los píxeles de la subdivisión pasarán a tener ese valor. Esto se realiza en forma independiente para cada color (rojo, verde y azul).

## I: Invertir

Al seleccionar esta acción, el programa debe cambiar el valor de cada píxel con la siguiente operación:

$$\text{Color}(i,j) = 255 - \text{Color}(i,j)$$

Esto se realiza en forma independiente para cada color (rojo, verde y azul).

## N: Blanco y negro (normal)

Al seleccionar esta acción, el programa debe cambiar el valor de cada píxel de esta forma:

$$\text{Si } (\text{Color}(i,j) \geq \text{promedio}) \text{ entonces: } \text{Color}(i,j) = 255$$

$$\text{en caso contrario: } \text{Color}(i,j) = 0$$

*promedio* es el promedio de los valores del color en toda la imagen. Esto se realiza en forma independiente para cada color (rojo, verde y azul).

## D: Blanco y negro (dither)

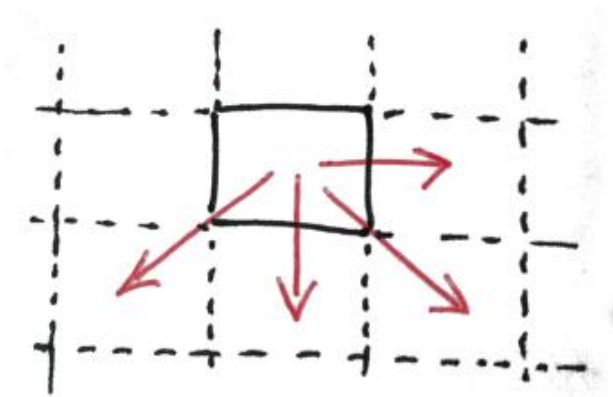
Al seleccionar esta acción, el programa debe cambiar el valor de cada pixel de esta forma

Si ( Color(i,j) >= promedio ) entonces: Color(i,j) = 255

en caso contrario: Color(i,j) = 0

*promedio* es el promedio de los valores del color en toda la imagen.

Después de hacer cada cambio, se debe calcular la diferencia entre el valor antiguo del color y su valor nuevo. Esta diferencia se debe distribuir (sumar) a los pixeles vecinos al píxel i,j de la siguiente forma:



$\frac{7}{16}$  al píxel de la derecha

$\frac{3}{16}$  al píxel de la izquierda y abajo

$\frac{5}{16}$  al píxel de la abajo

$\frac{1}{16}$  al píxel de la abajo a la derecha

Esto se realiza en forma independiente para cada color (rojo, verde y azul).

## M: Mostrar Imagen

Para mostrar la imagen, use el siguiente código:

```
1 import numpy as np
2 from matplotlib import pyplot as PLT
3
4 # ancho: el ancho de la imagen que queremos mostrar
5 # alto: el alto de la imagen que queremos mostrar
6 # r: una lista con los valores en el color rojo
7 # g: una lista con los valores en el color verde
8 # b: una lista con los valores en el color blue
9 #
10
11 def mostrarImagen(ancho, alto, r, g, b):
12     M = np.dstack((r, g, b))
13     PLT.imshow(M)
14     PLT.show()
```

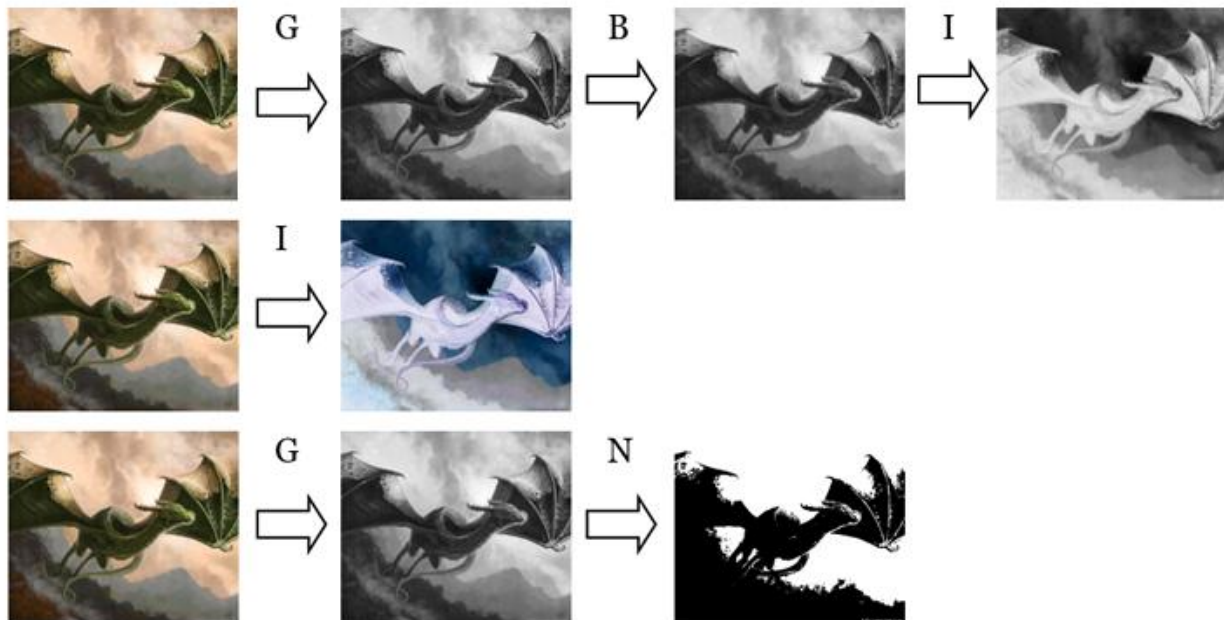
Un ejemplo de uso con ruido:

```

1 import numpy as np
2 from matplotlib import pyplot as PLT
3
4 # ancho: el ancho de la imagen que queremos mostrar
5 # alto: el alto de la imagen que queremos mostrar
6 # r: una lista con los valores en el color rojo
7 # g: una lista con los valores en el color verde
8 # b: una lista con los valores en el color blue
9 #
10 def mostrarImagen(ancho, alto, r, g, b):
11     M = np.dstack((r, g, b))
12     PLT.imshow(M)
13     PLT.show()
14
15 width = 400
16 height = 600
17
18 r = np.random.rand(height, width)
19 g = np.random.rand(height, width)
20 b = np.random.rand(height, width)
21
22 mostrarImagen(width, height, r, g, b)

```

## Ejemplos



Problema B. Ineludible (60%)

Siguiendo con el tema de la vialidad, se ha vuelto difícil controlar la velocidad a la que circulan los vehículos por las autopistas, los controles policiales son ineficientes y soluciones como radares estacionarios solo logran reducir la velocidad en el punto donde están ubicados y una vez cruzados, los conductores vuelven a acelerar. En países europeos existe hace varios años un sistema que es prácticamente invulnerable y que permite controlar la velocidad de los automovilistas en todo su trayecto. Se le conoce como control de velocidad por tramo o de velocidad media.

A diferencia de un radar estacionario, este sistema utiliza varios pórticos a lo largo de la ruta y al pasar por uno, este captura la patente del vehículo y la hora en que lo atravesó, luego cuando el conductor llega al siguiente pórtico, el sistema toma el tiempo y verifica cuánto demoró en recorrer el tramo. Si el conductor demora menos que el tiempo permitido, significa que andaba a una velocidad mayor que la permitida.

Por ejemplo, si entre el pórtico 1 y el pórtico 2 hay 100 km de distancia y la velocidad máxima del tramo es de 100 km/h, significa que cualquier vehículo que demore menos de una hora habrá viajado a una velocidad media superior a los 100 km/h, por lo tanto, será multado.

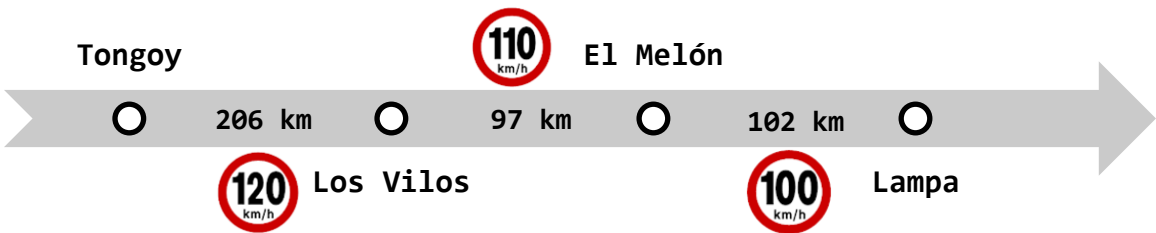
Se quiere implementar este sistema en toda la ruta 5, pero por el momento se probará un piloto entre Coquimbo y Santiago. La empresa encargada te pide a ti que crees el sistema que se encarga de las multas y de verificar los pagos de las multas. Se te entregará un archivo de texto llamado **radares.txt** con la siguiente estructura:

patente;pórtico;hora

- donde:
- patente:** es la matrícula del vehículo.
  - pórtico:** es el número del pórtico de control de velocidad.
  - hora:** es el tiempo marcado al atravesar el pórtico. En horas:minutos en formato de 24 h.

La ruta piloto tendrá los siguientes pórticos:

Número de Pórtico	Nombre	Distancia con pórtico anterior	Velocidad máxima desde pórtico
Pórtico 1	Tongoy	-	120 km/h
Pórtico 2	Los Vilos	206	110 km/h
Pórtico 3	El Melón	97	100 km/h
Pórtico 4	Lampa	102 km	-



Cada que un vehículo completa un tramo se verifica su velocidad media, y se aplica una multa según el tipo de falta. La multa se aplica en cada tramo que se exceda el límite.

Grave	Entre 10 y 20 km/h sobre el límite permitido.	1,5 UTM
Gravísima	Sobre 20 km/h	3 UTM

Además, se le entregará un archivo con las multas pagadas llamado **pagos.txt** que tendrá el siguiente formato:

patente, forma\_pago

donde:

**patente:** corresponde al vehículo infractor.

**forma\_pago:** corresponde a cómo fue pagada la multa total. Puede ser **ANTES** de la citación al juzgado, con lo que tendrá un descuento del 50% del valor, el día de la **CITACIÓN** pagando el valor completo o con **EXHORTO** que tiene un costo adicional de 20% por tramitación.

Según todo lo antes expuesto deberá calcular lo siguiente:

- 1) Mostrar los infractores, velocidad media y valor de la multa por tramo.
- 2) Multa total por patente.
- 3) Listado de infractores que no han pagado la multa.
- 4) Valor total recaudado sobre el total de multas cursadas y su porcentaje, en pesos chilenos.

Consideraciones:

- Ningún vehículo atravesará la ruta definida en 2 días. Todos los vehículos terminarán su ruta entre las 00:00 y las 23:59.
- Haga sus cálculos de velocidad redondeados a la unidad.
- Considere una UTM como 52.000 CLP.
- Considere que los vehículos solo van en dirección Los Vilos – Lampa y no en sentido contrario. Y que por lo tanto si un vehículo pasó por el pórtico X, la siguiente aparición del vehículo siempre será X+1.

radares.txt	Salida
AD-PIO-18;1;19:52 AE-IIF-92;1;9:44 BY-CPV-50;1;8:14 BZ-TTW-26;1;7:04 EK-BHH-53;1;18:17 ET-XBD-47;1;15:12 GQ-HNI-10;1;11:04 GV-EJA-18;1;8:22 HE-AMY-73;1;0:46 IW-ERG-33;1;5:13 KT-NOS-10;1;0:23 MI-LXU-84;1;6:04 ND-RIL-80;1;1:27 OH-PZY-15;1;18:59 OO-UDP-34;1;0:58 QC-IGA-39;1;0:51 QR-FRK-33;1;9:38 RC-NSD-12;1;12:48 RI-NT0-70;1;18:29 SC-UIA-47;1;11:53 SO-QAF-18;1;12:14 TJ-GRQ-06;1;18:57 UZ-UUJ-09;1;4:11 VT-XQH-37;1;2:16 ...	1) Infracciones BY-CPV-50 Tongoy - Los Vilos 242 km/h 3 UTM ET-XBD-47 Tongoy - Los Vilos 134 km/h 1.5 UTM GQ-HNI-10 Tongoy - Los Vilos 130 km/h 1.5 UTM KT-NOS-10 Tongoy - Los Vilos 325 km/h 3 UTM MI-LXU-84 Tongoy - Los Vilos 252 km/h 3 UTM OH-PZY-15 Tongoy - Los Vilos 145 km/h 3 UTM QC-IGA-39 Tongoy - Los Vilos 151 km/h 3 UTM QR-FRK-33 Tongoy - Los Vilos 136 km/h 1.5 UTM RC-NSD-12 Tongoy - Los Vilos 242 km/h 3 UTM TJ-GRQ-06 Tongoy - Los Vilos 167 km/h 3 UTM UZ-UUJ-09 Tongoy - Los Vilos 169 km/h 3 UTM VT-XQH-37 Tongoy - Los Vilos 179 km/h 3 UTM ET-XBD-47 Los Vilos - El Melón 121 km/h 1.5 UTM GQ-HNI-10 Los Vilos - El Melón 135 km/h 3 UTM KT-NOS-10 Los Vilos - El Melón 306 km/h 3 UTM MI-LXU-84 Los Vilos - El Melón 253 km/h 3 UTM NG-WVJ-60 Los Vilos - El Melón 162 km/h 3 UTM OH-PZY-15 Los Vilos - El Melón 132 km/h 3 UTM UI-DFP-80 Los Vilos - El Melón 253 km/h 3 UTM ET-XBD-47 El Melón - Lampa 120 km/h 1.5 UTM GQ-HNI-10 El Melón - Lampa 125 km/h 3 UTM KT-NOS-10 El Melón - Lampa 322 km/h 3 UTM MI-LXU-84 El Melón - Lampa 115 km/h 1.5 UTM OH-PZY-15 El Melón - Lampa 180 km/h 3 UTM QR-FRK-33 El Melón - Lampa 120 km/h 1.5 UTM RC-NSD-12 El Melón - Lampa 204 km/h 3 UTM VT-XQH-37 El Melón - Lampa 139 km/h 3 UTM
pagos.txt	2) Total por infractor BY-CPV-50 3 UTM ET-XBD-47 4.5 UTM GQ-HNI-10 7.5 UTM KT-NOS-10 9 UTM MI-LXU-84 7.5 UTM OH-PZY-15 9 UTM QC-IGA-39 3 UTM QR-FRK-33 3.0 UTM RC-NSD-12 6 UTM TJ-GRQ-06 3 UTM UZ-UUJ-09 3 UTM VT-XQH-37 6 UTM NG-WVJ-60 3 UTM UI-DFP-80 3 UTM
BY-CPV-50,CITACION ET-XBD-47,ANTES GQ-HNI-10,EXHORTO KT-NOS-10,EXHORTO MI-LXU-84,ANTES QC-IGA-39,EXHORTO QR-FRK-33,CITACION UZ-UUJ-09,ANTES VT-XQH-37,CITACION UI-DFP-80,ANTES	3) Recaudación \$2308800 recaudado de \$3666000 (62.98%) 4) Infractores morosos ['OH-PZY-15', 'RC-NSD-12', 'TJ-GRQ-06', 'NG-WVJ-60']