

Nombre y Apellido: _____

Tema 1 (15 pts.): El filtro de mediana ponderada asigna pesos individuales a las posiciones en la región de filtro, que puede interpretarse como el "número de votos" para los valores de los píxeles correspondientes. La distribución de pesos es especificado por una matriz de pesos $W(i,j) \in \mathbb{N}$. Para calcular el resultado del filtrado de una imagen I , cada pixel $(u,v) \in I$ es insertado $W(i,j)$ veces en un vector de píxeles extendido. Este vector es ordenado y el resultado del valor central es tomado como la mediana. La Figura 2 de abajo ilustra el cálculo del filtro mediana ponderada usando un filtro 3 x 3

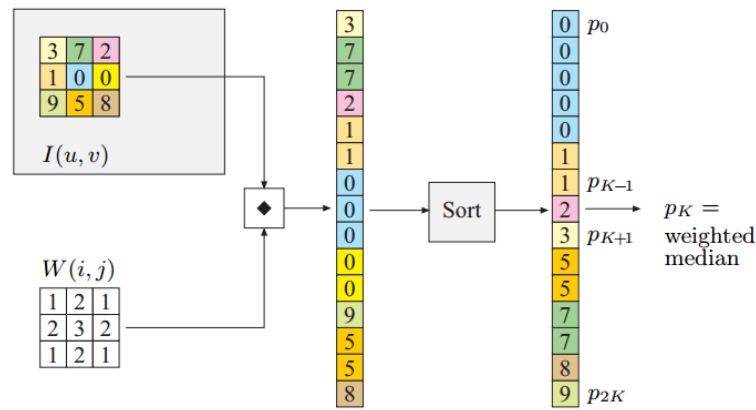
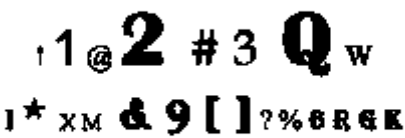


Figura 2 Filtro de la mediana ponderada

Programe una función en Matlab que reciba una imagen en escala de grises I , y una matriz de pesos W de tamaño $m \times n$, siendo m y n impares y el centro del filtro la posición $(m/2, n/2)$ y retorne una imagen J que sea el resultado de filtrar I utilizando el filtro W

```
function J = mediana(I,W)
```

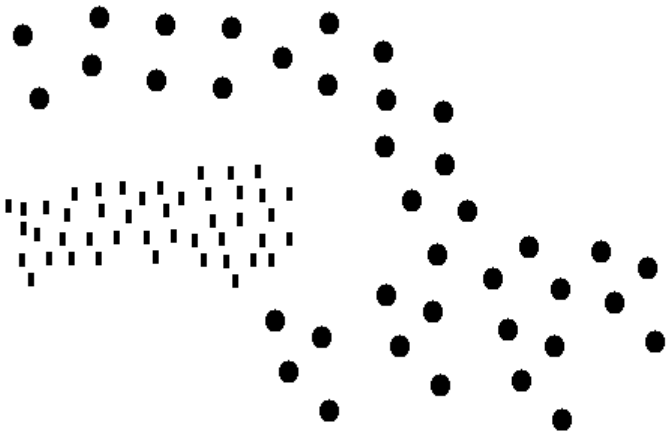
Tema 2 (15 pts) El objetivo de este ejercicio es generar, a partir de la imagen binumero.bmp, una nueva imagen que contenga sólo los dígitos 1,2 y 3. Generar un script en Matlab que solucione el ejercicio



Tema 3 (15 pts) Programar una función en Matlab que reciba una imagen binaria f y un valor n y que retorne una imagen g con los n objetos de mayor área. Se debe verificar que n no sea mayor a la cantidad de objetos existentes en la imagen

```
function g = mayorArea(f,n)
```







Tema 4 (15 pts) Programar una función en Matlab que dada la imagen de abajo consiga los píxeles que puedan servir de marcador de manera a poder reconstruir los objetos que no son circulares. Los marcadores se deben de conseguir utilizando la Transformada Hit-or-Miss y tablas de búsqueda. Queda estrictamente prohibido utilizar las funciones makelut y applylut de Matlab.



Tema 5 (20 pts) Mensajes antiguos

Para comprender las primeras civilizaciones, los arqueólogos estudian a menudo textos escritos en lenguas antiguas. Una de estas lenguas, utilizada en Egipto hace más de 3.000 años, se basa en caracteres llamados jeroglíficos. La Figura debajo muestra seis jeroglíficos y sus nombres. En este problema, escribirá una función en Matlab para reconocer estos seis caracteres.

```
function salida = jeroglíficos (imagen)
```

					
Ankh	Wedjat	Djed	Scarab	Was	Akeht

Entrada

La entrada describe una imagen que contiene uno o más jeroglíficos elegidos entre los que se muestran en la Figura. Las imágenes de entrada se ajustan a las siguientes reglas:

- La imagen contiene sólo los jeroglíficos mostrados en la Figura .
- Cada imagen contiene al menos un jeroglífico válido.
- Cada píxel negro de la imagen forma parte de un jeroglífico válido.
- Cada jeroglífico consiste en un conjunto conectado de píxeles negros y cada píxel negro tiene al menos otro píxel negro en su parte superior, inferior, izquierda o derecha.
- Los jeroglíficos no se tocan y no hay ningún jeroglífico dentro de otro jeroglífico.
- Los jeroglíficos pueden estar distorsionados, pero cada uno de ellos tiene una forma que es topológicamente equivalente a uno de los símbolos de la figura. (Dos figuras son topológicamente equivalentes si cada una puede ser transformada en el otro al estirarse sin quebrarse).

Salida

Una cadena que contenga un carácter por cada jeroglífico reconocido en la imagen, utilizando el siguiente código:

Ankh: A
Wedjat: J
Djed: D
Scarab: S
Was: W
Akhet: K

En cada cadena de salida, imprima los códigos en orden alfabético. Siga el formato de la salida de la muestra.

La entrada de muestra contiene descripciones de casos de prueba mostrados en las figuras 1 y 2.



Figura 1



Figura 2

Salida de ejemplo
Figura 1: AKW
Figura 2: AAAAAA

Tema 6 (20 pts) Programar una función en Matlab que reciba como parámetros una imagen binaria f y un valor n. Donde la imagen binaria f contiene figuras de polígonos de n vértices, donde de todos los vértices se desprende un línea recta a un mismo punto interior del mismo (Ver ejemplo de abajo) y retorne una imagen binaria g solo con los polígonos que tengan la cantidad de n vértices

```
function g = poligonos(f,n)
```

Polígonos ejemplos

