

Universidad Nacional de Asunción - Facultad Politécnica

Segundo Examen Final de Programación

Fecha: 04 de diciembre de 2021

Directivas y Recomendaciones

- El problema debe de ser resuelto usando el Lenguaje C y las herramientas vistas hasta la fecha del examen.
- Los comentarios aclaratorios pueden ayudar en la corrección del examen, influyendo favorablemente en su calificación.
- Utilice en sus programas los nombres de las estructuras indicadas en el enunciado.
- La duración del examen es de 120 minutos.
- El código fuente debe subirse al aula virtual de la materia (en EDUCA).
- La interpretación del problema es parte de la evaluación del examen.
- Solo si compila el ejercicio tendrá puntos y los mismos serán evaluados por casos de pruebas establecidos por el Profesor.
- El profesor estará evaluando los ejercicios con alguna herramienta de comparación de códigos. Códigos iguales o muy semejantes no serán considerados para la corrección

Ejercicio : Distancia entre pixeles

Una imagen binaria se puede representar como una matriz A de tamaño $M \times N$, donde cada celda de la matriz puede tener los valores 0 (negro) ó 1 (blanco). Tener en cuenta que la matriz tiene su origen en posición (1,1) recorriendo las filas de 1 a M y las columnas de 1 a N . Es decir, las coordenadas de la matriz A se puede ver como:

$$A = \begin{bmatrix} (1,1) & (1,2) & (1,3) & (1,4) \\ (2,1) & (2,2) & (2,3) & (2,4) \\ (3,1) & (3,2) & (3,3) & (3,4) \end{bmatrix}$$

El pixel en la fila “ x ” y en la columna “ y ” es llamada *pixel* (x,y). La distancia Euclidiana entre dos pixeles $p=(x1,y1)$ y $q=(x2,y2)$ se define como

$$d(p,q) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

Se pide realizar las siguientes tareas:

1. Leer la representación de una imagen binaria a partir de un archivo de entrada llamada **binaria.txt**.
2. Programar una función con nombre **transformadaDistancia** que reciba como parámetro un puntero a una matriz A (que representa una imagen binaria) y un puntero a una matriz B , donde cada valor de la celda de esta matriz tendrá la distancia Euclidiana al pixel blanco (con valor 1) más cercano del pixel en esa posición en A .
3. Guardar los resultados en un archivo **distancia.txt**.

Entrada del programa:

En la primera línea del archivo hay un par de enteros M y N separados por tabulador \t). ($1 \leq M \leq 10$, $1 \leq N \leq 10$). En cada una de las M siguientes líneas hay una cantidad N de enteros (cada entero puede tener un valor de 0 ó 1, separándose entre sí por tabulador \t).

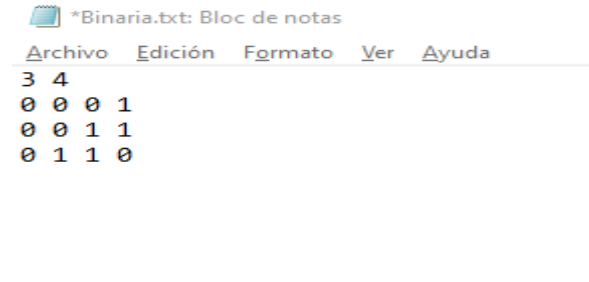
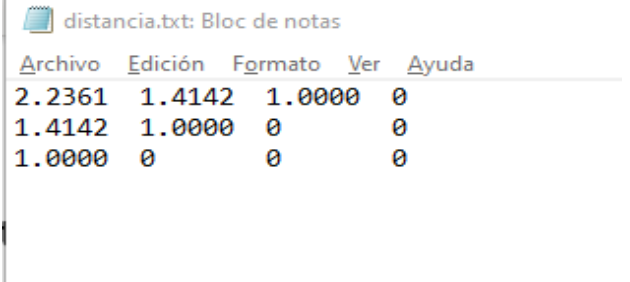
Validar:

1. Valor de M ($1 \leq M \leq 10$)
2. Valor de N ($1 \leq N \leq 10$).
3. Validar que los valores de las celdas sólo sean 0 ó 1

Salida del programa:

Un archivo donde en la línea x -ésima ($0 \leq x < M$) deben escribirse N enteros separados por tabulador (\t), con la distancia del pixel (x,y) al pixel blanco más cercano. Redondear a cuatro decimales el resultado del cálculo de la distancia.

Ejemplo:

Entrada del programa	Salida del programa
	

Nota: El archivo es solamente de ejemplo. El programa debería funcionar con cualquier entrada que cumpla con el formato descrito.

En el ejemplo para el pixel $p=(1,1)$ el pixel más cercano con valor 1 es el pixel $q=(2,3)$, por lo tanto:

$$d(p,q) = \sqrt{(1-2)^2 + (1-3)^2} = 2.2361$$

Valoración del Programa (Puntaje):

- Respetar el formato de las entradas y las salidas esperadas del programa (15 % del ejercicio).
- La salida de la función implementada (**transformadaDistancia**) es correcta (30 % del ejercicio).
- Utilizar aritmética de punteros para realizar operaciones (15 % del ejercicio).
- La salida del programa es correcta (40 % del ejercicio).