LABORATORIO N°3

CURSO DE MÉTODOS DE PROGRAMACIÓN

2-2020

UdeSantiago de Chile

CONTENIDO

1. Introducción

En este documento se muestra el enunciado del tercer laboratorio a desarrollar dentro del curso de Métodos de Programación, el cual consiste en encontrar una solución al acertijo llamado "*Nonogramas*". Este problema está dentro de los problemas en donde dado un conjunto de reglas dadas para una matriz de 2 dimensiones, al momento de cumplirlas, se forma una figura sobre ella. Este también ha recibido nombres distintos como *"pintando por números", "Picross"* o *"Pic-a-pix"*.

En este documento se explica el problema en sí, para posteriormente indicar qué es lo que deberá desarrollar usted en este laboratorio. A continuación, se muestran las reglas que posee este trabajo y su forma de evaluación. Finalmente se informan de las fechas, los elementos entregables, la forma de hacerlo llegar y otros elementos necesarios.

2. Contexto

Un grupo de programadores de alto conocimiento se encuentran trabajando en el desarrollo de una nueva tecnología correspondiente al renacimiento del uso de las tarjetas perforadas.

Para esto, se han dado cuenta que las tarjetas las podría construir el computador directamente, para lo cual ellos deberán darles solo las instrucciones en donde perforar. Cada tarjeta es una representación de una matriz, en donde cada cuadrante puede estar perforado o no, por lo que ellos le darán a la máquina la posibilidad de perforar la tarjeta, pero al ser un trabajo tedioso de estar indicando las coordenadas de dónde perforar o no, se han decidido basarse en los *Nonogramas* para realizar el conjunto de instrucciones, ya que es mucho más fácil para ellos.

Los *Nonogramas* es un tipo de puzzle que se basa en el colorear cierta parte acorde a ciertas reglas que se dan, como el mostrado en el la Figura 1, donde se da un conjunto de números los cuales indican cual es la cantidad de cuadros consecutivos pintados deben haber. Acorde a la figura, esta indica que:

- Filas:
  - Fila 1: No posee cuadros pintados
  - Fila 2: Posee 4 cuadros pintados consecutivamente
  - Fila 3: Posee 6 cuadros pintados consecutivamente
  - Fila 4: Posee 2 cuadros pintados consecutivamente, uno o más espacios en blanco y otros 2 cuadros pintados consecutivamente
  - Fila 5: Posee 2 cuadros pintados consecutivamente, uno o más espacios en blanco y otros 2 cuadros pintados consecutivamente

- Fila 6: Posee 6 cuadros pintados consecutivamente
- Fila 7: Posee 4 cuadros pintados consecutivamente
- Fila 8: Posee 2 cuadros pintados consecutivamente
- Fila 9: Posee 2 cuadros pintados consecutivamente
- Fila 10: Posee 2 cuadros pintados consecutivamente
- Fila 11: Posee 0 cuadros pintados consecutivamente

- Columnas:
  - Columna 1: Posee 0 cuadros pintados consecutivamente
  - Columna 2: Posee 9 cuadros pintados consecutivamente
  - Columna 3: Posee 9 cuadros pintados consecutivamente
  - Columna 4: Posee 2 cuadros pintados consecutivamente, uno o más espacios en blanco y otros 2 cuadros pintados consecutivamente
  - Columna 5: Posee 2 cuadros pintados consecutivamente, uno o más espacios en blanco y otros 2 cuadros pintados consecutivamente
  - Columna 6: Posee 4 cuadros pintados consecutivamente
  - Columna 7: Posee 4 cuadros pintados consecutivamente
  - Columna 8: Posee 0 cuadros pintados consecutivamente

					2 2			
		0	9	9	2 2	4	4	0
	0							
	4							
	6							
2	2							
2	2							
	6							
	4							
	2							
	2							
	2							
	0							

Figura 1: Nonograma vacío

De esta forma, siguiendo las indicaciones dadas una posible solución es la mostrada en la Figura 2.

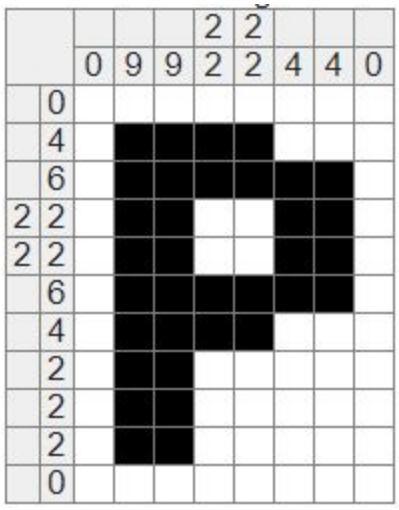

Figura 2: Nonograma completo

El problema que posee este grupo de programadores es que no les basta con solo generar la solución, sino que también necesitan saber si esta solución es única o no, con el fin de poder lograr instrucciones de programación de esta forma, y así acotar su nuevo lenguaje de programación de tarjetas perforadas.

3. Desarrollo de una solución

Para representar las tarjetas perforadas, se hará uso de una representación interna de una matriz, la cual será de tamaño 5x5 siempre. Mientras que las reglas serán dadas por dos archivos de texto, ingresado por el usuario, el cual siempre tendrá solo 5 números de dos dígitos, dónde las opciones pueden ser solamente:

- 00: Esa fila o columna no posee lugares de perforación.
- 01: Esa fila o columna posee solo un lugar de perforación.
- 02: Esa fila o columna posee dos lugares de perforación consecutivos solamente.
- 03: Esa fila o columna posee tres lugares de perforación consecutivos solamente.
- 04: Esa fila o columna posee cuatro lugares de perforación consecutivos solamente.
- 05: Esa fila o columna posee cinco lugares de perforación consecutivos solamente.
- 11: Esa fila o columna posee un lugar de perforación, luego un espacio o espacios sin perforar y posteriormente otro lugar de perforación.
- 12: Esa fila o columna posee un lugar de perforación, luego un espacio o espacios sin perforar y posteriormente otros dos lugares de perforación consecutivos.

- 13: Esa fila o columna posee un lugar de perforación, luego un espacio o espacios sin perforar y posteriormente otros tres lugares de perforación consecutivos.
- 21: Esa fila o columna posee dos lugares de perforación, luego un espacio o espacios sin perforar y posteriormente otro lugar de perforación.
- 22: Esa fila o columna posee dos lugares de perforación, luego un espacio o espacios sin perforar y posteriormente otros dos lugares de perforación consecutivos.
- 31: Esa fila o columna posee tres lugares de perforación, luego un espacio o espacios sin perforar y posteriormente otro lugar de perforación.

El primer archivo leído representará a las filas y el segundo a las columnas, y los nombres de estos archivos serán dados por el usuario.

Luego de tener las reglas del Nonograma a generar, el programa debe indicar si la instrucción ingresada es:

- Una instrucción única, indicando su solución.
- No es una instrucción única, indicando 2 posibles soluciones.
- No es posible representarla.

4. Trabajo a realizar

El trabajo a realizar consiste en crear un programa que permita encontrar la salida correspondiente al problema de las tarjetas perforadas del grupo de programadores, indicando si la configuración dada no posee solución, posee una solución o más de una.

Para esto, el programa deberá informar al usuario cual es o cuales son las soluciones que se poseen, según corresponda. Para esto es obligatorio que utilice el método de resolución de problemas Búsqueda en espacio de soluciones.

4.1. Características de entrada y salida

Dentro del desarrollo del trabajo se deben considerar explícitamente los siguientes elementos para su desarrollo.

Con respecto a las entradas de su problema el programa debe ser capaz de recibir la información descrita en el enunciado de parte del usuario por dos archivos de texto, cuyo nombre los dará el usuario y siempre estarán ubicados en una carpeta llamada “Pistas”. Este archivo de texto, tal como ya se explicó, siempre tendrá 5 números de dos

dígitos, dónde las opciones pueden ser solamente: 00, 01, 02, 03, 04, 05, 11, 12, 13, 21, 22 y 31, una opción por cada una de las filas, de esta forma un ejemplo de entradas son los mostrados en la Figura 3.

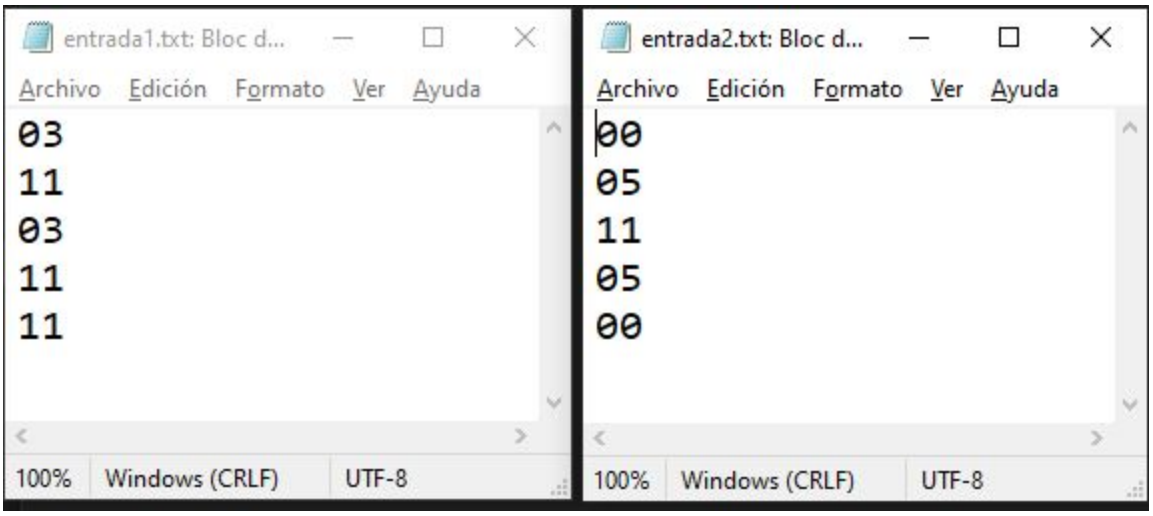

Figura 3: Ejemplo de entradas

Para la salida, esta también debe ser por salida estándar para el usuario, indicando claramente:

- Cuántas son las soluciones posibles
- Mostrar las soluciones, al menos dos siempre que se puedan mostrar estas dos.

De esta forma, la salida deseada para la solución indicada anteriormente sería:

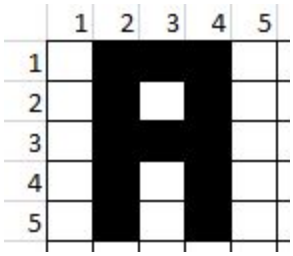

Figura 4: Solución al par de entrada dado en la Figura 3.

Pero en cambio, las entradas mostradas en la Figura 5, dan más de una solución, las cuales se muestran en la Figura 6.

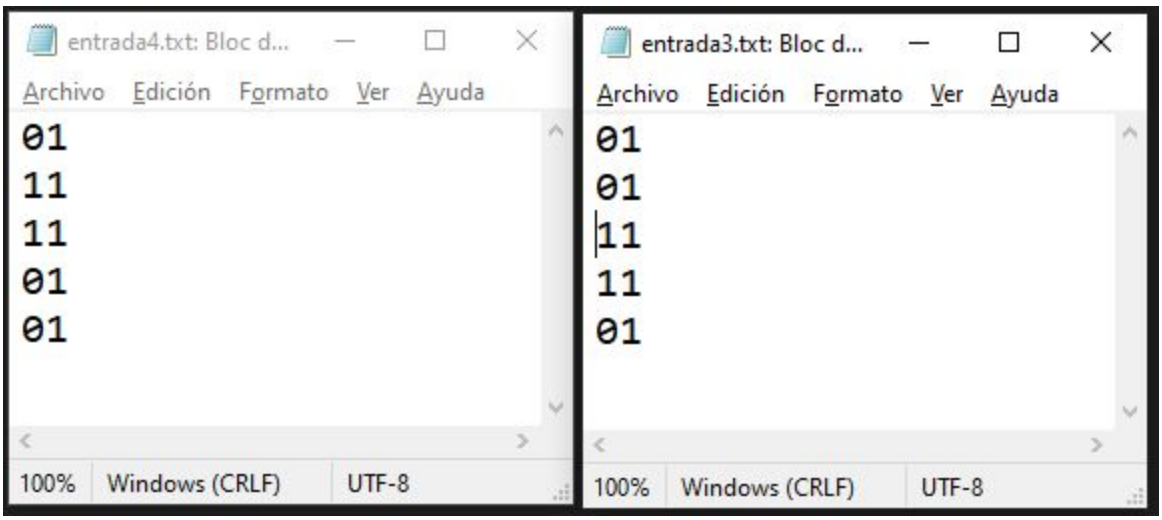

Figura 5: Ejemplo de entradas con más de una solución

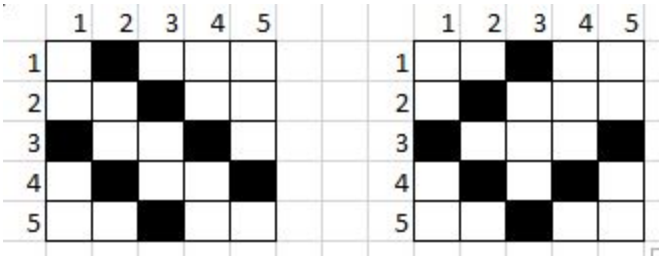

Figura 6: Posibles soluciones a la entrada de la Figura 5.

En caso de no haber solución, solo se debe indicar que no se encuentra solución para el par de instrucciones ingresadas.

4.2. Funcionalidades

Para el desarrollo del programa es necesario que se cumplan con el siguiente listado de funcionalidades:

1. El programa debe estar escrito en lenguaje de programación C.
2. El programa debe funcionar para Sistemas Operativos Unix, IOS y Windows.
3. La validación de la entrada.
4. El programa debe poseer un menú, en el cual solicite al usuario:
    a. Si desea ingresar un nuevo par de archivos a leer, siempre se debe pedir las reglas de las filas primero y luego las reglas de las columnas.

b. Salir

5. El programa debe informar al usuario cuando finalice su ejecución y debe volver a preguntar al usuario si desea probar nuevamente el programa.
6. El programa debe darle al usuario la salida por pantalla.
7. Para el proceso completo de ver el cuantas soluciones se poseen acorde al par de archivos de restricciones queda restringido a utilizar Búsqueda en Espacio de Soluciones, pero se debe justificar dentro del documento readme el uso de si ésta. Además en el mismo Readme debe indicar si el conjunto de soluciones iniciales se realiza mediante iteración o recursión.

5. Entregas y evaluación

La entrega consiste en la subida al portal del curso de Métodos de Programación, en el sitio www.udesantiagovirtual.cl, en la actividad Entrega Laboratorio Individual N°1 el día **01 de marzo del año 2021 a las 23:55 horas**.

En específico se deberá entregar un archivo comprimido en formato .zip o .rar que:

- Que tenga como nombre la siguiente estructura **<Sección Laboratorio>_<RUN>**:
  - Primero debe ir la sección del laboratorio, por ejemplo, L23.
  - Luego el RUN del creador del código, por ejemplo, 12345678-9
  - Entonces el nombre final será L23_12345678-9
- El código fuente del programa en extensión .c
- Un archivo de texto plano llamado README, el cual debe indicar un paso a paso el cómo funciona su programa.
  - Este es un mini manual de usuario, explicando a grandes rasgos el cómo funcionan ciertos elementos, por ejemplo, si hay un menú, indicar si las opciones se ingresan de forma numérica, palabras, si estas van en mayúsculas o minúsculas, etc. Si el programa tiene que leer o escribir o interactuar con un archivo de texto, dónde debe estar ubicado. Si la interacción con el usuario es mediante entrada y salida estándar o mediante alguna interfaz gráfica.

Para la evaluación, se revisará en específico los siguientes puntos, con nota de 0 a 6 para cada uno, y el porcentaje correspondiente a la ponderación que tendrá cada punto,

el cual está indicado entre paréntesis, posteriormente se sumará el punto base a su evaluación:

- Entrada y salida de forma correcta y específica al enunciado (15%).
- Uso de menús (5%).
- Indicación del paso a paso para encontrar la solución al problema utilizando Búsqueda en espacio de soluciones. (30 %)
- Justificación e implementación del método respectivo indicado en el readme y coherencia con lo implementado. (10 %)
- Uso de buenas prácticas, en dónde (10%):
    - El código fuente debe tener como comentario, al inicio de él, la siguiente información:
        - RUN del creador del código.
        - Sección del laboratorio a la cual pertenece el creador del código.
        - Fecha de creación del código.
    - Las funciones creadas deben indicar:
        - Entradas: Qué significa cada una de las entradas de las funciones, por ejemplo, si se tiene una función suma que recibe dos números enteros, indicar que la entrada corresponde a los operandos de la operación adición.
        - Salida: Qué significa la salida generada, y en caso de tener salidas que representen valores del tipo Verdadero o Falso, opciones 1, 2, 3, …, n; indicar qué significa que salga cada una de éstas.
        - Objetivo de la función.
    - Uso de nombres de variables representativos.
    - Uso de identación y orden del código.
- Pruebas (30 %):
    - La validez de las prueba no consistirá solamente en mostrar o dar el resultado solicitado, sino que también la aplicación de los métodos de resolución de problemas solicitados, en caso de aplicar el método solicitado el puntaje es válido, incluso cuando el método sea aplicado de forma errónea. En caso de no ver la aplicación de los métodos solicitados se considerará que no son pruebas válidas, por lo que serán evaluadas con puntaje 0.