

ESTRUCTURA DE DATOS

TRABAJO FINAL DE SEMESTRE

Consideremos una población de **K** bacterias distribuidas en una matriz de dimensiones **M × N**, de modo que en cada una de las celdas hay, como máximo, una bacteria. Además, consideramos que, las 8 bacterias en las celdas que rodean a una bacteria son sus **vecinas** (tanto las horizontales, verticales y diagonales).

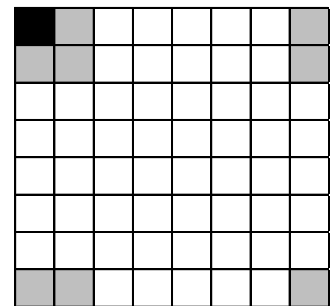
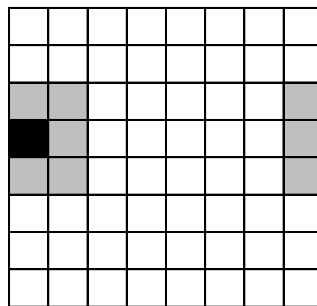
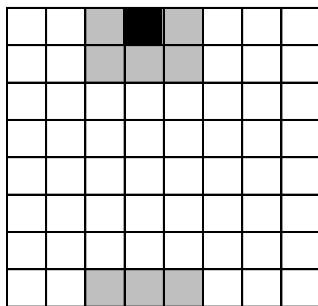
Las bacterias pueden tener 2 estados posibles: viva o muerta. El ciclo de vida de estas bacterias sigue las siguientes reglas de evolución:

- Una bacteria muerta **renace** si está rodeada con exactamente **3 bacterias vecinas vivas**.
- Una bacteria viva con **2 o 3** bacterias vecinas **vivas** sigue con vida.
- En cualquier otro caso, una bacteria **muere o permanece muerta**.

Cada una de estas reglas, se evalúan en ciclos. Un ciclo es una evaluación de la matriz completa donde se decide qué bacterias renacen, mueren o se mantienen en el mismo estado.

Considere que:

- Todas las celdas en los extremos de la matriz se consideran vecinos con sus opuestos, por ejemplo, según la siguiente imagen, los vecinos de la celda negra son todas las celdas grises.



- Las bacterias que nacen o mueren no afectan a las reglas hasta que se ha completado un ciclo evolutivo. Esto quiere decir que al ir calculando los nuevos estados de las celdas de la matriz las modificaciones se deben hacer de manera auxiliar (por ejemplo, utilizando una estructura de datos diferente) para luego reflejarla en la matriz final.

I. Usted debe escribir un programa que realice lo siguiente:

1. Abra un archivo que describe un estado inicial de la matriz de bacterias (la estructura del archivo será explicada más abajo).
2. Muestre por pantalla el estado actual de la matriz de bacterias. Usted elige el formato en el que se mostrará el estado de vida o muerte de cada bacteria, pero se debe reconocer claramente la estructura de la matriz.
3. Simule la evolución de la población durante **L** ciclos evolutivos.
4. Guarde en un archivo el estado actual de la matriz en el mismo formato que el archivo de entrada que se pide en el punto número 1.
5. **Debe** hacer uso de las estructuras de dato vistas en clase para la solución de los ejercicios, por ejemplo, utilice una Pila para almacenar los cambios de estado y después reflejarlos en la matriz original.

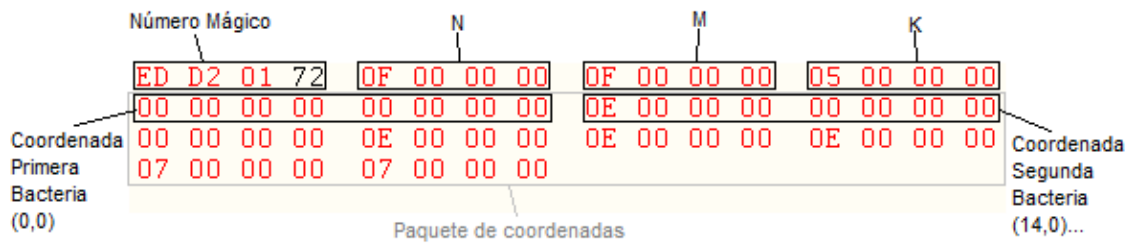
II. Debe presentar un informe que incluya lo siguiente:

1. Resumen de la estrategia que utilizó para desarrollar el programa.
2. Estructuras de datos utilizadas para resolver el problema, indicando:
 - a. Sus atributos.
 - b. Sus operaciones.
 - c. Sus axiomas.

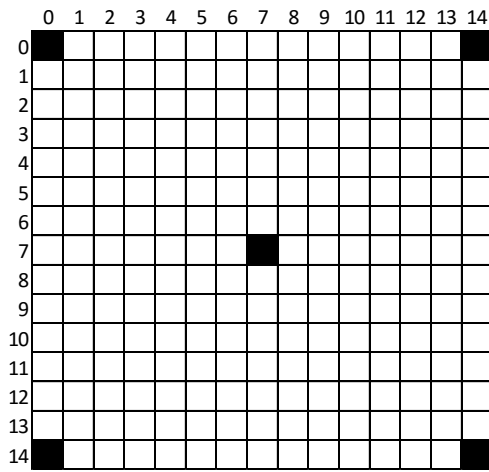
La estructura del archivo debe ser la siguiente:

- Debe ser un archivo **binario**.
- Debe iniciar con los siguientes 4 bytes: **0xEDD20172**, al abrir el archivo debe validar este número mágico, en caso de que no aparezca debe indicarlo con un error.
- En la cabecera del archivo continuaría con 2 números enteros (4 bytes cada uno) que representan N y M respectivamente.
- Continúa con un número entero (4 bytes) representando el número **K**, para una cierta cantidad de bacterias a ser descritas a continuación.
- Después de esto viene un paquete de **K** descriptores de bacterias vivas que serán puestas en la matriz. Cada descriptor incluirá 2 números enteros (4 bytes cada uno) que representan las coordenadas (x,y) en la matriz de cada bacteria viva en el estado inicial de la matriz.
- Todos los números en el archivo, exceptuando el número mágico son almacenados utilizando el formato Little-endian.

A continuación, se presenta un ejemplo de archivo que podría recibir.



Este archivo representaría la siguiente matriz:



Donde las celdas negras representan una bacteria viva y las blancas una muerta.

Observaciones:

- Todos los programas deben ser desarrollados mediante el uso de funciones que serán llamados desde el programa principal.
- Los ejercicios serán desarrollados en lenguaje C.
- La realización del taller es en grupo de máximo 2 personas.
- Únicamente pueden trabajar en equipo 2 personas que asistan activamente a un mismo laboratorio, no se evaluarán proyectos que no cumplan este requisito.
- El código deberá ser subido al repositorio GitHub de todos los integrantes.
- La copia será evaluada con la nota mínima.

La fecha de entrega será el día **Miércoles 22 de Noviembre a las 23:59 hrs.**