

TRABAJO PRÁCTICO N° 4

Deben entregarse por grupos en la entrega correspondiente vía Campus los archivos .c correspondientes a los ejercicios indicados.

1. Escribir una función que calcule el producto de una matriz de tres columnas por un vector de tres filas. El prototipo de la función requerida es el siguiente:

```
void prm3(double vr[], double M[][3], double v[], int filas);
```

donde filas es el número de filas de la matriz M. La función realiza el producto de la matriz M por el vector v y el resultado lo coloca en el vector vr. La matriz y la cantidad de filas deben ser ingresadas por teclado.

2. Escribir una función que calcule el determinante de una matriz de 2x2.
3. Escribir una función que calcule el determinante de una matriz de 3x3.
4. Escribir una función que calcule la inversa de una matriz de 3x3.

5. [ENTREGAR] Juego de la vida

[El juego de la vida](#), inventado en 1970 por John Conway in 1970, es un simulador de células autómatas. El juego consiste de un mundo 2D que se extiende infinitamente en todas las direcciones, dividido en celdas que representan una célula. Cada célula puede estar únicamente en uno de dos estados: viva o muerta, en cada una de las generaciones. El juego consiste de una serie de reglas que describen cómo estas células evolucionan de generación en generación.

Estas reglas calculan el estado de una célula para la próxima generación como función del estado de las células colindantes (vecinas) en la generación actual. En un mundo de dos dimensiones, estas son las 8 celdas que se encuentran vertical, horizontal o diagonalmente adyacentes a la celda. Las reglas se pueden resumir como:

- a) Una célula viva con menos de dos vecinos muere;
- b) una célula con más de tres vecinos también muere;
- c) una célula viva con exactamente dos o tres vecinos sobrevive;
- d) una célula muerta con exactamente tres vecinos obtiene vida nuevamente.

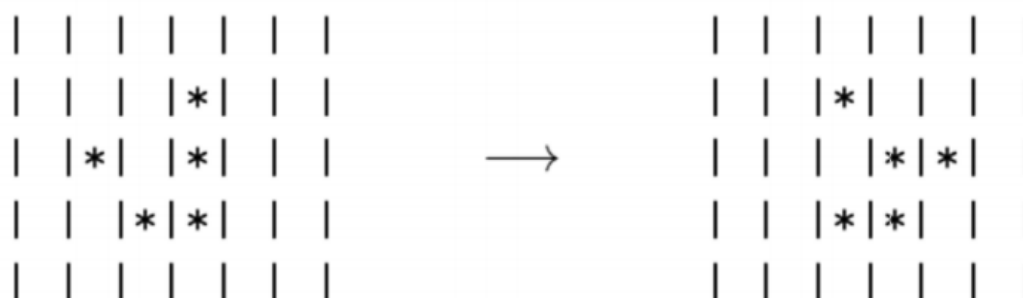


Figura 1: Ejemplo

Para este ejercicio, vamos a implementar el Juego de la vida, con la restricción menor de que nuestro mundo es finito. A su vez, asumiremos que los vecinos más allá de los límites de nuestro mundo se encuentran muertos (y nunca son actualizados). El estado inicial estará hardcoded (es decir, estará fijado en el código fuente del programa).

Para el desarrollo del mismo se brinda a continuación una guía de funciones que pueden resultarles útiles para organizar el desarrollo (no es obligatorio utilizarlas, y pueden ser modificadas también).

```
/*
 * Life Game
 * Funciones de acceso al mundo
 * ¡El objetivo de las mismas es que sean independientes de las
 * reglas del juego!
 */

/* initialize_world -- set up world, all cells initialized
 * to DEAD or ALIVE; all cells in next generation are
 * initialized to DEAD
 */
void initialize_world(void);

/* returns the width (x) and height (y) of the world
 */
int get_world_width(void);
int get_world_height(void);

/* returns the state (DEAD or ALIVE) of the cell at (x,y);
 * coordinates go from x = 0,...,width-1 and
 * y = 0,...,height-1; returns DEAD for cells outside this
 * range
 */
int get_cell_state(int x, int y);

/* sets the state (DEAD or ALIVE) of the cell at (x,y) in
 * the next generation; range of coordinates same as in
 * get_cell_state()
 */
void set_cell_state(int x, int y, int state);

/* updates world state to next generation and resets all
 * next generation states to DEAD
 */
void finalize_evolution(void);

/* outputs the current world state to the console
 */
void output_world(void);
```

Algunos requisitos del programa a realizar:

- a) El tamaño del mundo debe encontrarse en dos constantes, por ejemplo, ANCHO y ALTO. Debe poder cambiarse el valor de las mismas y recompilar el código, y el mismo debe funcionar.
- b) El programa debe mostrar el estado actual del mundo en pantalla, y luego preguntar al usuario cuantas generaciones desea avanzar. Cuando el usuario ingresa un número y presiona enter, se calcula el nuevo estado del mundo, se presenta en pantalla, y se vuelve a preguntar.
- c) Si el usuario presiona enter sin ingresar ningún número, se debe avanzar una generación.
- d) Si el usuario ingresa el carácter 'q', se debe salir del programa.
- e) Cualquier funcionalidad extra que deseen agregar se tomará en cuenta en la nota del trabajo práctico.