

El MONOCROM es un monitor monocromo con pantalla cuadrada formada por una retícula de 16x16 puntos, cada uno de los cuales se puede poner de color blanco o de color negro.

El monitor dibuja a partir de ciertas órdenes que recibe en forma de secuencia de caracteres; los caracteres enviados actúan no sobre toda la pantalla, sino solamente sobre un área seleccionada, que puede cambiarse mediante algunos de estos caracteres.

Inicialmente, el área seleccionada es toda la pantalla, y puede suponerse que todos sus puntos son de color blanco. El carácter 'x' sirve para pintar el área seleccionada (es decir, poner todos sus puntos de color negro) e, inmediatamente, volver a considerar como área seleccionada la pantalla entera.

Finalmente, los caracteres del '1' al '4' sirven para hacer más pequeña el área seleccionada, de manera que posteriores caracteres 'x' afecten a menos puntos. Concretamente, estos caracteres escogen como nueva área seleccionada uno de los cuatro trozos siguientes:

'1': trozo superior izquierdo

'2': trozo superior derecho

'3': trozo inferior derecho

'4': trozo inferior izquierdo

Nótese, pues, que en realidad MONOCROM sólo permite pintar uno o más cuadrados; cualquier figura compleja debe pintarse a trozos, seleccionando y pintando los cuadrados adecuados.

Así, por ejemplo, la siguiente secuencia:

```
2 2 2 2 x
3 x
1 1 3 x
```

Debería crear lo siguiente:

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 x
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 x x 0 0 0 0 0 0 0 0 0 0 0 0
0 0 x x 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 x x x x x x x x
0 0 0 0 0 0 0 0 x x x x x x x x
0 0 0 0 0 0 0 0 x x x x x x x x
0 0 0 0 0 0 0 0 x x x x x x x x
0 0 0 0 0 0 0 0 x x x x x x x x
0 0 0 0 0 0 0 0 x x x x x x x x
0 0 0 0 0 0 0 0 x x x x x x x x
0 0 0 0 0 0 0 0 x x x x x x x x
```

## Ejercicio 1

Escribe un proyecto Node.js con las funciones necesarias para nuestra aplicación:

- Función que inicializa la pantalla
- Función que recibe una orden y pinte los cuadrados adecuados
- Función que imprime la pantalla por consola

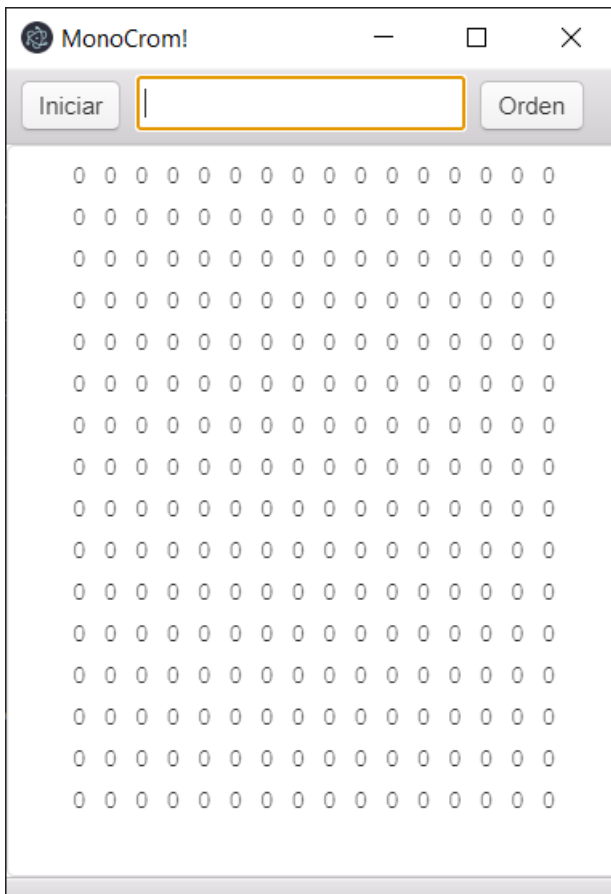
Incluye un código que reproduzca las órdenes anteriores y muestre el resultado por pantalla

Un posible punto de partida es este. Ten en cuenta que la matrix se divide en 4 cuadrantes y ese cuadrante a su vez en otros 4 cuadrantes y así sucesivamente:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

## Ejercicio 2

Crea una interfaz utilizando un *textarea* como pantalla. Deberías obtener algo parecido a esto:



En este ejemplo el formato del *textarea* no es puramente Photon. Está modificado con este estilo:

```
<style type="text/css">
  textarea {
    resize: none;
    font-family: Courier;
    text-align: center}
</style>
```