

Ejercicio 1 (6 ptos)

Vives en una ciudad, donde todas las calles están dispuestas en una cuadrícula perfecta.

Llegas unos minutos antes a una cita, así que decides aprovechar la oportunidad para dar un corto paseo.

La ciudad ofrece a sus ciudadanos una aplicación para móviles (en realidad la haremos en JavaScript), **generadora de paseos**: cada vez que la ejecutas, proporcionas un número (los minutos que tienes libres) y recibes una secuencia de letras que representan la dirección en la que te debes dirigir durante cada minuto ('n' de norte, 's' de sur, 'o' de oeste, 'e' de este).

Siempre caminas solo una manzana por cada letra (dirección) y sabes que te lleva un minuto atravesar una manzana de la ciudad.

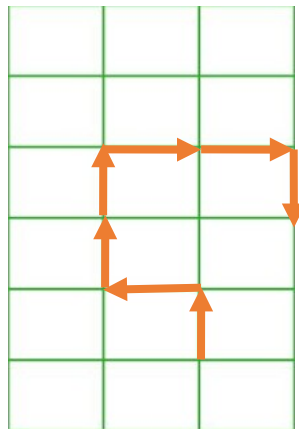
Crea una función tal que, proporcionándole el tiempo del que dispones en minutos, obtenga una **cadena** de letras (tantas como minutos dispongas) que te permitirá pasear y volver al punto inicial exactamente en los minutos introducidos (¡no quieres ni llegar temprano ni retrasarte!).

Nota: siempre recibirás “en la pantalla de tu dispositivo” una **matriz** no vacía válida que contiene una secuencia aleatoria de letras de dirección. Por ejemplo, para diez minutos, podría ser: ["n", "e", "n", "o", "s", "o", "s", "s", "e", "n"]

La aplicación debe funcionar solo para tiempos comprendidos entre 8 y 22 minutos

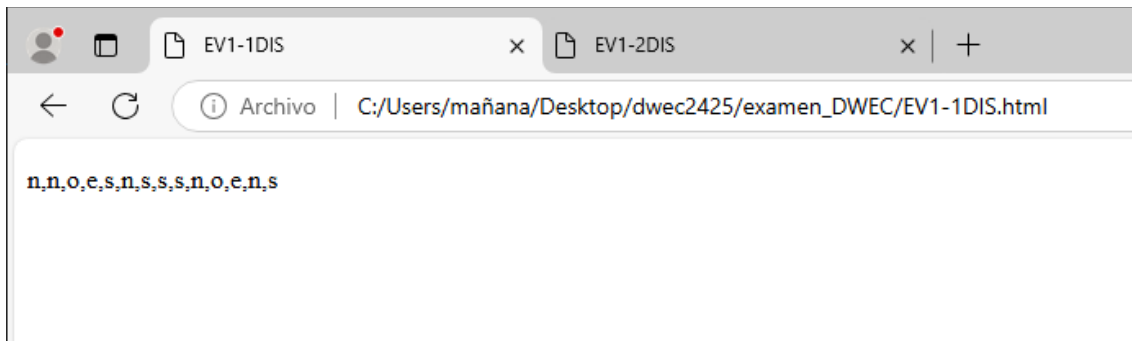
Simula el resultado con una petición de 14 y añade una captura del resultado obtenido por pantalla.

En el dibujo adjunto se representa gráficamente la secuencia (no válida) “nonnees”



Calificación (sobre 6 puntos)

- Funcionamiento (2,5 puntos)
- Generación de números aleatorios, (Math.floor(Math.random())) redondeo y asignación a punto cardinal (n,s,e,o) y creación de la secuencia (1 pto)
- Control de errores (1,5 ptos)
- Utilización de variables, constantes y funciones con camelCase y descriptivas. Comentarios (0,5 ptos)
- Transformación cadena-matriz y aspecto gráfico de la información proporcionada al usuario (0,5 ptos)



Ejercicio 2 (4 ptos)

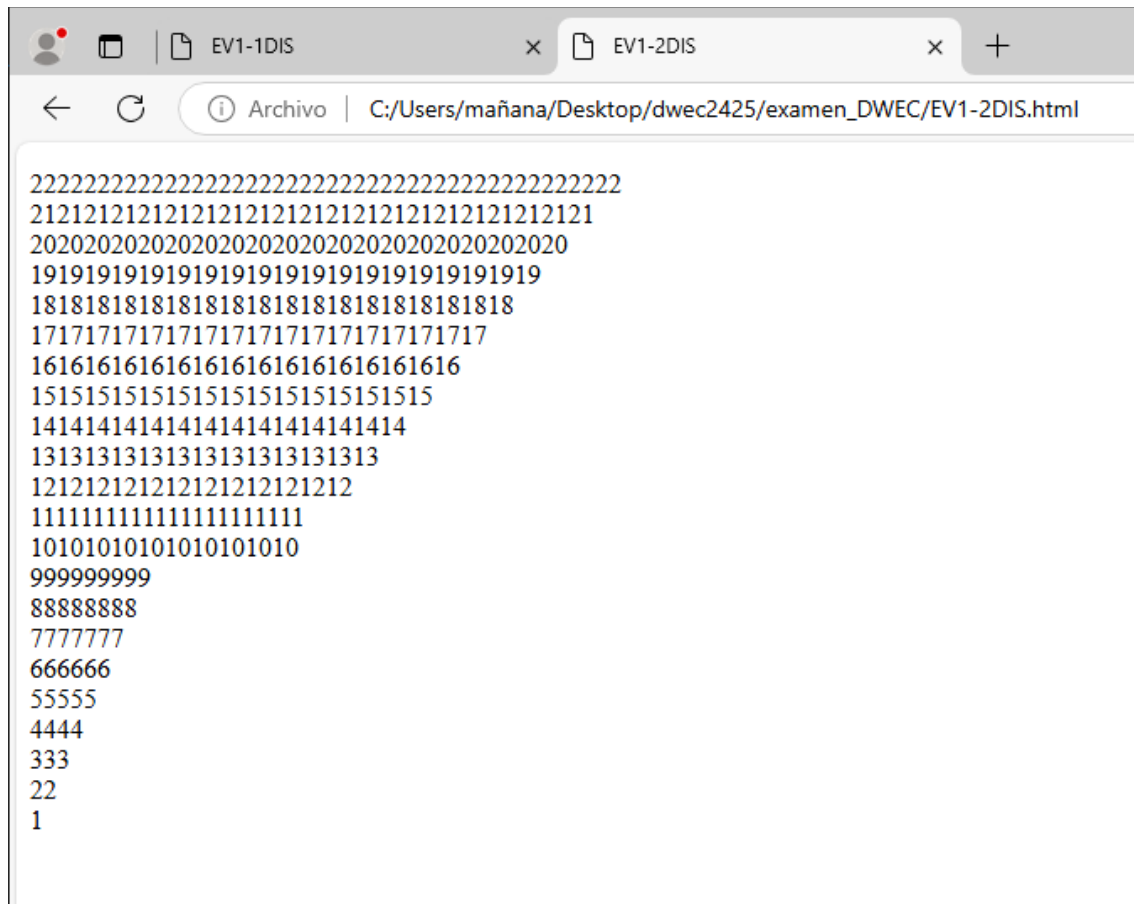
Haz un programa en JavaScript que dibuje una pirámide invertida utilizando los números del 1 **al número que indique el usuario** (máximo 50), de tal forma que la primera línea contenga tantos números repetidos como el número introducido y vaya disminuyendo en una unidad en cada salto de línea. Las cifras estarán centradas en la pantalla en la medida de lo posible. Simula el resultado si el número indicado es 22 y añade una captura del resultado obtenido por pantalla.

Ejemplo. Si se indica el número 30 el resultado sería:

Línea 1:	30303030303030303.....30	(El número 30, 30 veces)
Línea 2:	292929292929292...29292929292929292929292929292929	(El número 29, 29 veces)
Línea 3:	2828282828...28282828282828282828282828282828	(El número 28, 28 veces)
	
Línea 28:	333	(El número 3, 3 veces)
Línea 29:	22	(El número 2, 2 veces)
Línea 30:	1	(El número 1, 1 vez)

Calificación (sobre 4 puntos)

- Funcionamiento (1,75 puntos)
- Bucles (0,75 pts)
- Control errores (0,75 pts)
- Utilización de variables, constantes y funciones con camelCase y descriptivas. Comentarios (0,50 pts)
- Aspecto gráfico (centralidad) (0,25 pts)



Instrucciones generales

- Aunque no se trata de una buena práctica, por necesidades del sistema de corrección, **el código JS debe ir embebido en el HTML**.
- Sube al aula virtual **dos ficheros HTML** , sin comprimir, utilizando el formato de nombre:
EV1-1XXX.html y **EV1-2XXX.html** donde XXX son tus tres iniciales habituales.
- Sube, además, este enunciado con el nombre **EV1-XXX.pdf** y añade en cada apartado una captura del resultado obtenido para N=22.