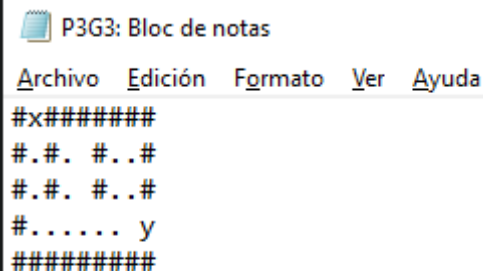


Práctica Backtracking

- Resolver los problemas utilizando algoritmos de backtracking
 - Problema 1: horse walk Etiqueta - P1
 - Problema 2: n-queens Etiqueta - P2
 - Problema 3: maze Etiqueta - P3
- Los problemas 1 y 2
 - Entrada n= tamaño del lado del tablero (nxn), x, y que indica la pos. inicial (comenzando desde 0,0)
 - Estos problemas se ejecutan tres veces
- El problema tres
 - Todos los grupos crean su propio laberinto, el laberinto que sea mas difícil tiene 2 puntos extra sobre toda la práctica
 - Entrada es un archivo de texto plano
 - paredes # caminos posibles "." (un punto), los bordes tienen pared #, la entrada tiene la letra x, y la salida la letra y, ambos en un borde, no tiene espacios.
 - Sobre el algoritmo, a priori solo sabe donde esta x, no conoce y
 - [Ejemplo](#)



```
P3G3: Bloc de notas
Archivo Edición Formato Ver Ayuda
#x#####
#.#. #..#
#.#. #..#
#..... y
#####
```

- Todos los grupos corren los laberintos de todos y se suma el tiempo.
- El algoritmo tiene que correr para todos los laberintos.
- La impresión será, con tab

#	x	#	#	#	#	#	#	#
#	1	#	7	#	.	.	.	#
#	2	#	8	#	.	.	.	#
#	3	4	9	10	11	12	13	y
#	#	#	#	#	#	#	#	#

- Salida general toda función retorna solo un tiempo
 - PIG1(in)
 - ini = time()
 - ...
 - Impresión
 - fin = time()
 - ->> double
 - Para el problema 2
 - puntos los vacíos, x las reinas, espaciado por tab
 - Para el problema 1
 - espaciado por tab, sin bordes, mostrando la numeración
- Los tiempos de cada ejecución se suman y es lo que retorna cada función
 - Los nombres de los archivos tienen el siguiente formato: horse_1.h donde el número indica el número id del grupo, dentro de la carpeta horse de C1, el nombre de la función sería horse_1 y retorna el tiempo de ejecución.
 - [C1]
 - Drive: https://drive.google.com/drive/folders/14KDZ8LCJ3x_qqKfA-YM5axEcFINITH5Y?usp=sharing
- No threads
- Ejecución
 - time1 = horse_1(n,x,y);
- Sobre los grupos
 - "Max(Tres participantes)"
 - Colocar en el spreadsheet los integrantes G1,G2...
- Como medir el tiempo
 - <https://www.delftstack.com/es/howto/cpp/how-to-get-time-in-milliseconds-cpp/>
- Calificación
 - Solo hacerlos **correr** vale 2 puntos por cada problema (6pts)

- Los otros catorce puntos dependen del puesto y del número de grupos
- P1 G1- 2 G2 -1.56 G3 - 3.002 G2 recibe 3 puntos. G1 recibe 2 y G3 recibe 1