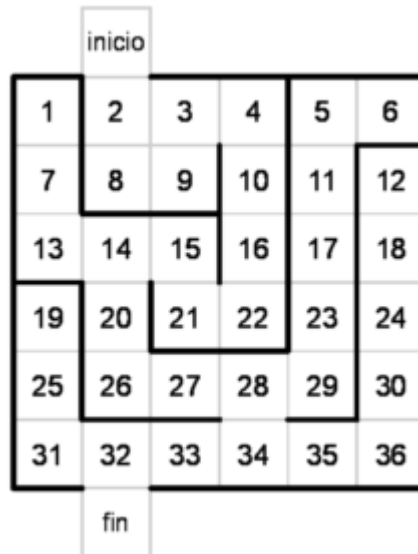


## Práctica 2 de Lenguajes – Paradigma Funcional

### Ejercicios

#### NOTAS:

- Los siguientes ejercicios pretenden que el programador se enfoque en el algoritmo que resuelve el problema y no en la interacción con el usuario para obtener datos, así que se pueden trabajar con datos quemados y haciendo pruebas de funciones/métodos controladas por el programador en el main de los programas si se requiere.
  - Los ejercicios deben ser implementados por el/la estudiante y no utilizar otras funciones puntuales que existan que lo realicen ya en F# (si es que existe)
  - Solo el primer ejercicio es necesario subirlo al repositorio
- 
- 1) Modifique el ejercicio de búsqueda en profundidad visto en clase para cálculo de rutas en un grafo, para ingresar pesos a los vértices y con esta nueva información, calcular el camino más corto en la búsqueda en profundidad basándose en la sumatoria de pesos de los vértices de la ruta. Una opción sin mucho cambio al código hecho es generar todos los caminos posibles (con su información de pesos) y luego a partir de esto, filtrar el de menos tamaño en función a la información de pesos.
  - 2) Construya un programa que resuelva el problema de encontrar la o las rutas de un laberinto con datos quemados. Dicho problema se puede resolver exactamente igual que un problema de grafos, entendiendo que cada casilla del laberinto puede verse como un “vecino” adjunto de otra, siempre que no haya pared:



Así por ejemplo, en el laberinto anterior, se puede entender que por ejemplo, la posición 2 está conectada con la 3 y la 8 y en ese sentido y siguiendo el ejemplo de búsqueda en profundidad visto en clase, se puede representar como el vértice  $[[2],[3,8]]$  del eventual grafo que representa todo el laberinto.

- Represente el laberinto del gráfico como un grafo y corra la búsqueda en profundidad para encontrar todos los caminos que lo resuelven. Elimine paredes y represente ese nuevo conocimiento de caminos posibles en el grafo y vuelva a probar.
- Escriba una función que encuentre la ruta más corta para resolver el laberinto (entendiendo rutas más cortas en este problema, no como un tema de pesos, si no como caminos más cortos en cantidad de pasos – entre más rápido llegue a la solución, mejor)