

ICOM2020 – 2do Parcial

18 de diciembre de 2020

Notas:

1. Uso de prácticos: **se pueden utilizar los trabajos prácticos propios realizados.**

Problema 1: Estructuras de información de cátedras

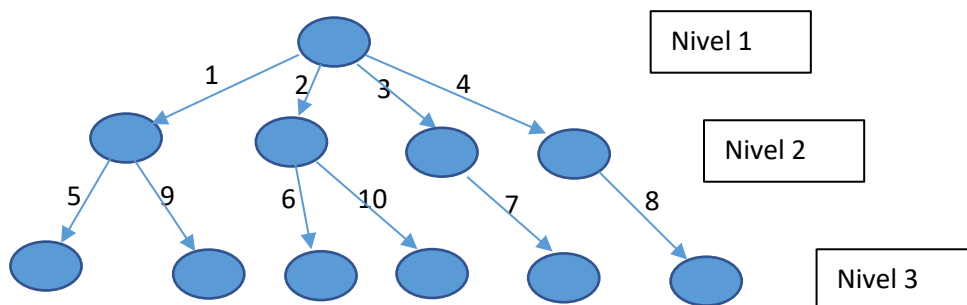
Se desea implementar una versión simplificada de un sistema para llevar los datos de cursadas de alumnos.

En el esqueleto de código `cursada.cpp` se encuentra una versión preliminar del diseño de clases con una implementación parcial de las mismas.

Complete la implementación y pruébela a través del `main` presente en el código.

Problema 2: Árboles n-arios

Un árbol n-ario es un árbol en donde cada nodo puede tener 0 o más hijos (que cuelgan de él) que son a la vez nodos (árboles n-arios).



Se pretende implementar un árbol n-ario con algunos requerimientos especiales:

- a. La cantidad máxima de nodos hijos de un nodo debe ser 4.
- b. El agregado de nodos a un árbol se realice de forma que el peso de los nodos (número de nodos que penden de él) sea lo más balanceado posible.
- c. Se crearán nuevos niveles a medida en que los niveles anteriores no puedan contener más nodos por el requerimiento a.

Utilizando el esqueleto de `nnode.cpp` en donde hay una definición parcial de la clase `NNode`, complete los métodos faltantes.

Problema 3: Calabozos y Neutrones

Felicitaciones, ha sido contratado como el nuevo oficial de radioprotección de una antigua instalación supersecreta de algún gobierno. El director de la instalación, que también es nuevo, como todo el personal, quiere encomendarle que diseñe los planes de evacuación de la instalación ante distintos escenarios de accidentes. En el archivo `mapa.txt` encontrará los planos de las instalaciones. Como primer accidente se postula el caso (meramente hipotético) de una liberación en todo el complejo de aerosoles radiactivos. Aquellos empleados que estén lejos de la salida (marcada como **S** en el mapa) se refugiarán en el bunker (marcado como **B** en el mapa) que cuenta con un suministro independiente pero limitado de oxígeno. Cuando los niveles de radiación sean tolerables (o se acabe el oxígeno) deberá conducir al personal hasta la salida. Distribuidos por todo el complejo se encuentran detectores de radiación. Usando el archivo

niveles_radiacion.txt como ejemplo de la información provista por estos sensores, implemente un algoritmo que busque y resalte en el mapa el camino de mínima exposición a la radiación. Utilice como punto de partida el código dejado por su predecesor en el archivo laberinto_radiactivo.cpp. Esperamos que lo haga mejor que él...

```
#####
#           #   #   #   B#   #   #
# #####   #   #   #   #   #   #   #   #   #   #   #   #
# #       #   #       #   #   #   #   #   #   #   #   #
# # #   #####   #   #   #####   #####   #   #   #####
# # #   #   #   #   #   #   #   #   #   #   #   #   #
# # #   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
#### #   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
# #####   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
#####   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
# #####   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
# #####   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
# #####   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
# #####   #   #   #   #   #   #   #   #   #   #   #   #
#   #   #   #   #   #   #   #   #   #   #   #   #   #
#####
```