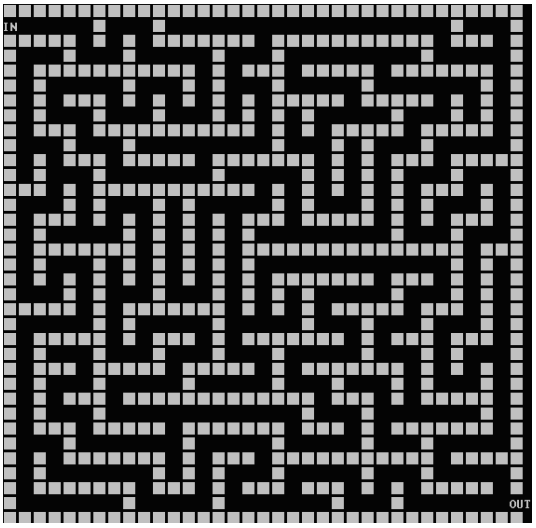
Projet – Programmation parallèle
Résolution d'un labyrinthe par *backtracking* avec collecte d'objets



Dans le cadre de ce projet, vous devez développer un algorithme permettant de résoudre un labyrinthe, en utilisant une approche de type *backtracking*. Le labyrinthe est représenté par une grille de caractères, dans laquelle différentes entités sont présentes.

L'objectif du programme est de trouver un chemin allant du point de départ D au point d'arrivée A, en passant par les trois objets mentionnés (couronne, bouclier et épée respectivement pour C, B, E).

L'ensemble des entités présents dans le labyrinthe seront expliqués lors du premier TP.

Vous devez écrire un programme en C++ qui permet de résoudre un labyrinthe via un algorithme de type *backtracking*. Votre programme doit prendre en entrée un fichier texte décrivant le labyrinthe et devra afficher le chemin de cette partie.

Travail à réaliser

- **Algorithme**
  - Implémenter un algorithme de *backtracking* séquentielle
  - Implémenter un algorithme de *backtracking* parallèles en utilisant la librairie thread de C++11. L'algorithme peut être paralléliser à

différents niveaux, vous devez implémenter au minimum deux variantes.
    - La fonction doit explorer les chemins possibles à partir de D.
    - Le chemin est valide uniquement s'il atteint A en ayant ramassé les trois objets.
- **Affichage**
    - Affichez le chemin trouvé (par exemple avec des *).
    - Affichez un message si aucun chemin valide n'existe.

Rendu du travail

Vous devrez fournir les code sources en C++ du projet ainsi qu'un rapport. Ce rapport devra contenir :

1. Un manuel d'utilisation de votre programme
2. Les explications concernant vos différentes implémentations des algorithmes de *backtracking*

Le rapport ne devra pas excéder 3 pages.

Il est fortement conseillé de prendre le temps de la réflexion avant d'arrêter un choix pour vos structures de données. En particulier, les structures de données choisies devront permettre une mise en œuvre aisée et efficace de résolution de *backtracking*.

Durant la phase de débogage, je vous invite à compiler en utilisation les options –Wall et –g3 du compilateur g++ et d'utiliser alternativement un débogueur comme gdb et un programme vérifiant les accès à la mémoire comme valgrind.

Une fois votre programme au point, il est conseillé de le compiler avec l'option d'optimisation –O3 du compilateur g++ afin d'obtenir un exécutable le plus performant possible. Dans le cas de figure où vos expérimentations sont effectuées sur un ordinateur portable, il est important de s'assurer que la politique de gestion de l'énergie est réglée sur la puissance maximum afin de ne pas biaiser les comparaisons.

Attention : de nombreuses ressources et programmes (de qualité variable) sont disponibles très facilement, notamment sur Internet. Vous pouvez bien évidemment les consulter, mais gardez bien à l'idée que c'est un travail personnel : l'enseignant responsable sera particulièrement vigilant sur ce point et prendra les sanctions adéquates en cas de détection.