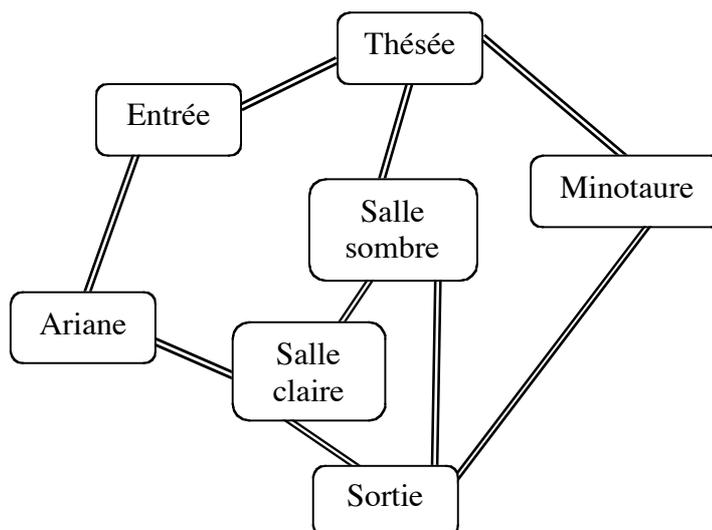


## TP numéro 4

### 1 Labyrinthe

- Modéliser le plan suivant en indiquant à Prolog les liens entre les salles.

*Indication* : on pourra utiliser `couloir(entree,thesee)` pour le premier lien.



- Écrire le prédicat `chemin(X,Y, Parcours)`, qui donne sous forme de liste le parcours qui permet d'aller de la salle X à la salle Y.
- Tester `chemin(entree, sortie, Parcours)`. Quels sont les problèmes ?
- Modifier le prédicat `chemin(X,Y,Parcours)` en `chemin_sans_boucle(X, Y, Parcours, ListeTaboue)` pour obtenir les solutions.
- Quelle requête permet de sortir du labyrinthe avec Ariane en évitant la salle du Minotaure ?

### 2 Coupure

- Définir trois versions du prédicat `soustrait(L1,L2,L3)`, qui étant données les listes L1 et L2, construit la liste L3 qui contient les éléments de L1 qui n'appartiennent pas à L2 :
  - avec un test,
  - avec une coupure,
  - en enlevant la coupure.

Comparer le fonctionnement de ces trois versions.

### 3 Négation

Définir un prédicat `disjoints(L1,L2)` satisfait si L1 et L2 sont deux listes qui n'ont aucun élément en commun (une ligne).

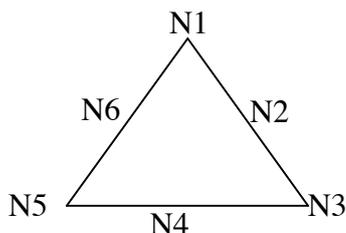
## 4 Génération

Définir le prédicat `permute(L1,L2)`, qui étant donnée la liste `L1`, construit la liste `L2` contenant une permutation de `L1`.

Exemple : `findall(L,permute([1,2,3],L),R)`.

`R = [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]`

## 5 Triangle de nombres



On veut placer les nombres de 1 à 6 en triangle, sur les sommets et sur les milieux des segments, en utilisant un nombre une fois seulement, de façon à ce que la somme des trois nombres utilisés soit la même pour les trois cotés du triangle.

1. En utilisant `permute`, écrire le programme qui donne la solution de ce problème.
2. Combien y a-t-il de solutions ? Le prédicat `length(X,Long)`, combiné à `findall` peut servir à les compter.

## 6 Pour aller plus loin (exercices facultatifs)

• Soit un programme contenant des faits tels que : `jeune(alfred)`, `jeune(toto)`, etc.

Ajouter à ce programme la règle suivante : « Si un individu est jeune, il aime le sport, sinon il aime le jardinage », en utilisant une coupure. On souhaite utiliser ce prédicat pour savoir ce qu'aime quelqu'un. Testez ce prédicat.

Utilisez le prédicat pour poser la question suivante : « Est-ce que toto aime le jardinage ? » Corriger votre programme.

• On considère le prédicat `insere(X,L1,L2)`, qui étant donnés un nombre `X` et une liste triée `L1`, construit la liste triée `L2` résultant de l'insertion de `X` dans `L1`.

Définir deux versions de ce prédicat, la première sans coupure, et la deuxième avec une coupure pour remplacer un test.

Comparer le comportement de ces deux versions en test et en génération. Que répond Prolog aux questions suivantes : `insere(8,[3,10],[8,3,10])`, `insere(8,[3,10],[3,10,8])`. Corriger votre programme.