

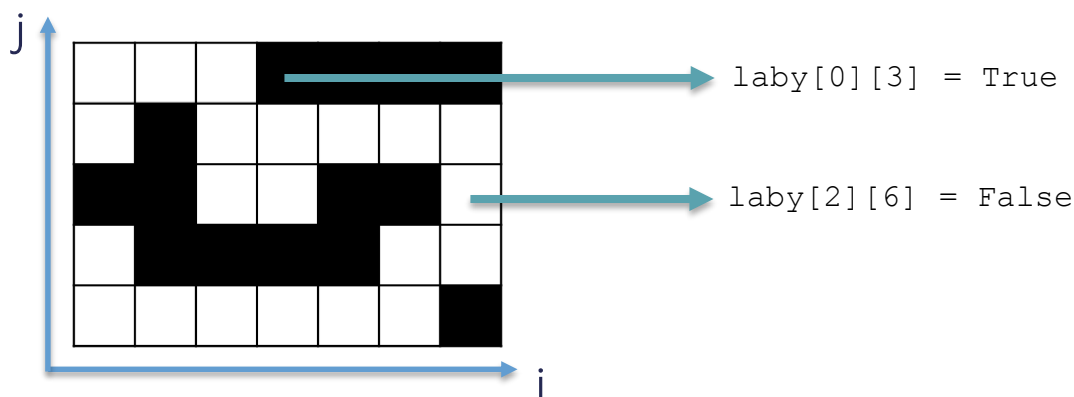
Labyrinthes

1) Labyrinthes simples

a) Représentation

Pour se familiariser avec les structures multidimensionnelles, on va dans un premier temps utiliser la représentation "simple" suivante :

- Un labyrinthe est représenté par un tableau de tableaux
- Chaque case contient un booléen
- Ce booléen répond à la question "*Cette case contient-elle un mur ?*"



b) Premiers labyrinthes

Question 1. Ecrire une fonction qui prend en argument deux entiers l et h , et qui renvoie une grille correspondant à un labyrinthe vide de largeur l et de hauteur h .

Question 2.a. Ecrire une fonction qui prend en argument un labyrinthe et qui renvoie sa hauteur.

Question 2.b. Ecrire une fonction qui prend en argument un labyrinthe et qui renvoie sa largeur.

c) Génération aléatoire

Question 3. Ecrire une fonction qui prend en argument trois entiers l , h et r , et renvoie une grille correspondant à un labyrinthe aléatoire de largeur l et de hauteur h , où chaque case a r % de chances de contenir un mur.

Indice : Pour créer un labyrinthe où chaque case a r % de chances de contenir un mur, il vous suffit de choisir pour chaque case un nombre aléatoire entre 1 et 100 : si ce nombre est inférieur ou égal à r , la case contient un mur.

d) Affichage

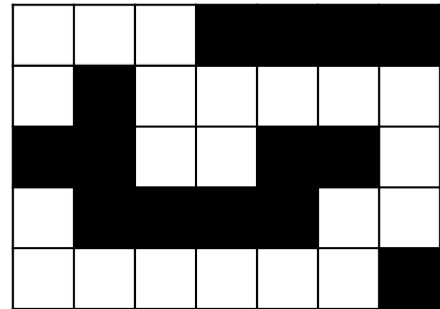
Question 4. Créer une fonction qui prend en argument un labyrinthe et qui utilise les caractères dièse (#) et espace () pour afficher ce labyrinthe

Exemple : Le labyrinthe ci-contre serait affiché de la manière suivante

```

    ###
#
##  ##
####
#

```



★ **Question 5.** Imaginer une fonction qui prend en argument un labyrinthe et qui utilise plusieurs caractères pour afficher ce labyrinthe de façon plus élaborée.

Exemple : Par exemple, le labyrinthe ci-dessus pourrait être représenté de la façon suivante :

```

+---+---+---+---+---+---+
|   |   |   |###|###|###|###|
+---+---+---+---+---+---+
|   |###|   |   |   |   |   |
+---+---+---+---+---+---+
|###|###|   |   |###|###|   |
+---+---+---+---+---+---+
|   |###|###|###|###|   |   |
+---+---+---+---+---+---+
|   |   |   |   |   |   |###|
+---+---+---+---+---+---+

```

e) Voisinage

Question 6. Ecrire une fonction qui détermine les "voisins accessibles" d'une case dans un labyrinthe, c'est-à-dire une fonction qui prend en argument un labyrinthe et les coordonnées *i* et *j* d'une case de ce labyrinthe, et qui renvoie la liste des cases voisines non occupées par des murs.

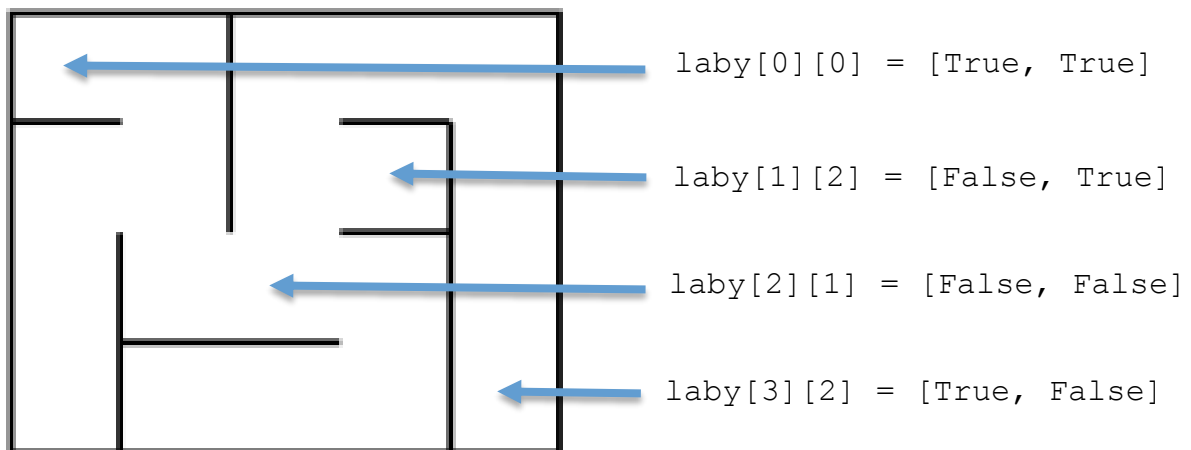
Exemple : Les voisins accessibles de la case (0, 1) du labyrinthe ci-dessous sont [(0, 0), (0, 2)].

2) Labyrinthes avancés

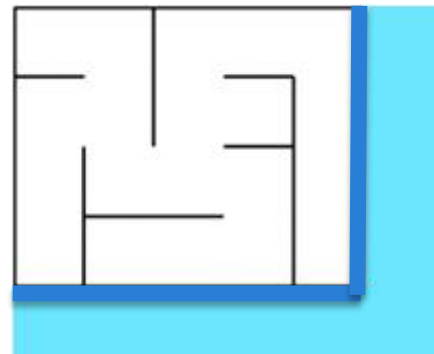
a) Représentation

Pour obtenir des labyrinthes à la fois plus esthétiques et plus intéressants, nous allons adopter une nouvelle représentation :

- Un labyrinthe est représenté par un tableau de tableaux
- Chaque case contient un tableau de deux booléens, qui répondent aux questions
 - "Est-ce qu'il y a un mur à gauche de cette case ?"
 - "Est-ce qu'il y a un mur au-dessus de cette case ?"



Remarque : Pour représenter un tel labyrinthe avec deux booléens par case, une solution simple consiste à ajouter une colonne et une ligne supplémentaire (ces cases "fictives" ne servent qu'à fermer le bas et le côté droit du labyrinthe)



b) Création d'un labyrinthe vide

Question 7. Ecrire une fonction qui prend en argument deux entiers l et h , et qui renvoie une grille correspondant à un labyrinthe vide de largeur l et de hauteur h .

Remarque : Les seuls murs à créer sont donc les murs extérieurs.

Question 8.a. Ecrire une fonction qui prend en argument un labyrinthe et qui renvoie sa hauteur (sans compter la ligne supplémentaire).

Question 8.b. Ecrire une fonction qui prend en argument un labyrinthe et qui renvoie sa largeur (sans compter la colonne supplémentaire).

c) Génération aléatoire

Question 9. Ecrire une fonction qui prend en argument trois entiers l , h et r , et renvoie une grille correspondant à un labyrinthe aléatoire de largeur l et de hauteur h , où pour chaque case il y a r % de chances d'avoir un mur sur la gauche, et r % de chances d'avoir un mur au-dessus.

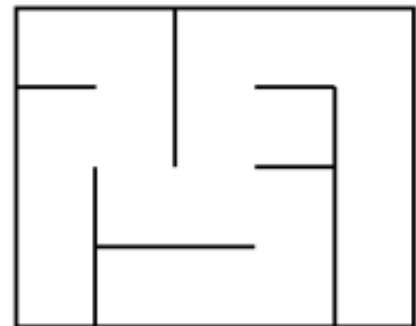
Indice : La méthode est similaire à celle utilisée à la question 3, sauf qu'on tire ici deux nombres aléatoires pour chaque case.

d) Affichage

Question 10. Créer une fonction qui prend en argument un labyrinthe et qui utilise plusieurs caractères pour afficher ce labyrinthe

Exemple : Par exemple, le labyrinthe ci-contre pourrait être représenté de la manière suivante

```
+---+---+---+---+
|       |       |
+---+ + +---+ +
|       |       |
+ + + +---+ +
| |       | |
+ +---+---+ + +
| |       | |
+---+---+---+---+
```



e) Voisinage

Question 11. Ecrire une fonction qui détermine les "voisins accessibles" d'une case dans un labyrinthe, c'est-à-dire une fonction qui prend en argument un labyrinthe et les coordonnées i et j d'une case de ce labyrinthe, et qui renvoie la liste des cases voisines non séparées par un mur.

Exemple : Les voisins accessibles de la case $(0, 1)$ du labyrinthe ci-dessous sont $[(0, 0), (1, 1)]$.