

# ET 6 Correction (éléments)

## Exercice :

1

```
def mur(l,i,j):  
    return l[i][j]==1
```

2a

La variable d représente le carré de la distance entre les cases *case1* et *case2*. Deux cases adjacentes ont une distance égale à 1. Donc  $d==1$  (et donc la fonction *voisine*) renvoie True si *case1* et *case2* sont adjacentes et False dans le cas contraire.

2b

```
def adjacentes(l):  
    adj = True  
    n = len(l)  
    for i in range(n-1):  
        if not voisine(l[i], l[i+1]):  
            adj = False  
    return adj
```

3

Avant le début de la boucle while la variable  $i = 0$ . À chaque itération de la boucle la variable  $i$  est incrémentée d'une unité ( $i = i + 1$ ). La boucle va "s'arrêter" quand  $i = \text{len}(\text{cases})$ . Comme le tableau *cases* n'est pas infini, la boucle va donc obligatoirement se terminer.

4

```
def echappe(cases, laby):  
    nb_ligne = len(laby)  
    nb_colonne = len(laby[0])  
    return cases[0] == (0,0) and cases[len(cases)-1] == (nb_ligne-1,  
nb_colonne-1) and teste(cases, laby)
```