

Exercice 4

Partie A

1

```
lab2[1][0] = 2
```

2

```
def est_valide(i,j,n,m):  
    return i>=0 and j>=0 and i<n and j<m
```

3

```
def depart(lab):  
    n = len(lab)  
    m = len(lab[0])  
    for i in range(n):  
        for j in range(m):  
            if lab[i][j]==2: return (i,j)
```

4

```
nb_cases_vides(lab):  
    n = len(lab)  
    m = len(lab[0]) compt = 0  
    for i in range(n):  
        for j in range(m):  
            if lab[i][j]==2 or lab[i][j]==3 or lab[i][j]==0:  
                compt = compt + 1 return compt
```

Partie B

1

L'appel de la fonction renvoie : [(2, 2), (1, 1)]

2a

```
# entrée: (1, 0), sortie (1, 5)  
chemin = [(1, 0)]  
chemin.append((1,1))  
chemin.append((2,1))  
chemin.pop()  
chemin.append((1,2))  
chemin.append((1,3))  
chemin.append((2,3))  
chemin.append((3,3))  
chemin.append((3,4))  
chemin.pop()  
chemin.pop()  
chemin.pop()  
chemin.append((1,4))  
chemin.append((1,5))
```

2b

```
solution(lab):
    chemin = [depart(lab)]
    case = chemin[0]
    i = case[0]
    j = case[1]
    while lab[i][j] != 3:
        lab[i][j]=4
        v = voisines(i,j,lab)
        if len(v) != 0 :
            prochaine = v.pop()
            chemin.append(prochaine)
            i = prochaine[0]
            j = prochaine[1]
        else :
            chemin.pop()
            n = len(chemin)
            i = chemin[n-1][0]
            j = chemin[n-1][1]

    return chemin
```