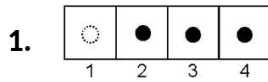
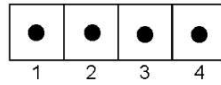


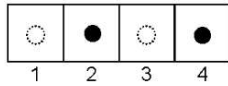
Exercice 2 : Jeu du baguenaudier



Il est possible de jouer la case 1 :



ou la case 3 :



2.

```
def initialiser(n):
    tab = []
    for i in range(n):
        tab.append(False)
    return tab
```

Autre solution :

```
def initialiser(n):
    return [False]*n
```

3.

```
def victoire(tab):
    for etat_case in tab:
        if etat_case == False:
            return False
    return True
```

4.

```
def indice_premiere_case_occupee(tab):
    for i in range(len(tab)):
        if tab[i]:
            return i
    return None
```

5.

```
def coup_valide(tab, case):
    if case==0:
        return True
    elif case>=0 and case<len(tab) and case==indice_premiere_case_occupee(tab)+1:
        return True
    else:
        return False
```

6.

```
def changer_case(tab, case):
    if coup_valide(tab, case):
        tab[case] = not tab[case]
    return tab
```

7. Principe : Pour remplir un baguenaudier de taille n , il faut partir d'un baguenaudier rempli de taille $n-1$ puis le vider jusqu'à la taille $n-2$ afin que seule la dernière case $n-1$ soit remplie et que l'on puisse remplir la case n .

```
def vider(n):
    if n == 1:
        print('Vider case 1')
    elif n > 1:
        vider(n-2)
        print(f'Vider case {n}')
        remplir(n-2)
        vider(n-1)
```

8.

La pile d'appel récursif est :

```
vider(3)
    vider(1)
        "Vider case 1"
    "Vider case 3"
    remplir(1)
        "Remplir case 1"
    vider(2)
        vider(0)
        "Vider case 2"
        remplir(0)
        vider(1)
            "Vider case 1"
```

L'affichage obtenu en console est :

```
Vider case 1
Vider case 3
Remplir case 1
Vider case 2
Vider case 1
```

9.

```
def remplir(n):
    if n == 1:
        print('Remplir case 1')
    elif n > 1:
        remplir(n-1)
        vider(n-2)
        print(f'Remplir case {n}')
        remplir(n-2)
```

10.

Pour un baguenaudier de 2000 cases, le nombre d'appels récursifs va exploser et dépasser la limite de la pile d'appel de 1000. Une erreur de type **RecursionError** arrêtera alors le programme.