

Interro — Récursivité**Calculatrice interdite****— 40'****Exercice 1** (3 points)

On considère la fonction Python `mystere` suivante qui prend un tableau d'entiers `t` et un entier `i` en paramètres.

```
def mystere(t, i):
    if i == 0:
        return True
    else:
        return t[i-1] < t[i] and mystere(t, i-1)
```

- Quel est le résultat de `mystere([1, 9, 11, 42], 2)` ? Dessiner l'arbre des appels correspondants.
- En supposant que `i` est toujours compris entre 0 et `len(t)`, expliquer en une phrase ce que fait la fonction `mystere(t, i)`.

Exercice 2 (4 points)

On appelle fonction récursive de Ackermann, la fonction `Ack` qui pour deux entiers naturels m et n associe l'entier $Ack(m, n)$ défini par :

$$Ack(m, n) = \begin{cases} n + 1 & \text{si } m = 0 \\ Ack(m - 1, 1) & \text{si } m > 0 \text{ et } n = 0 \\ Ack(m - 1, Ack(m, n - 1)) & \text{si } m > 0 \text{ et } n > 0 \end{cases}$$

- Calculer à la main :

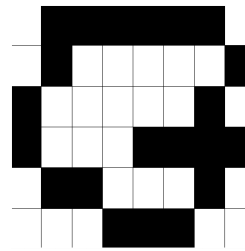
▷ $Ack(0, 1)$ ▷ $Ack(0, 2)$ ▷ $Ack(1, 0)$ ▷ $Ack(2, 0)$ ▷ $Ack(1, 2)$

Aide : penser à réinvestir les résultats trouvés, en particulier pour $Ack(1, 2)$

- Implémenter en python, la fonction `Ack(m, n)` correspondante.

Exercice 3 (3 points)

On représente une image par un tableau de chaînes de caractères. Chaque caractère représente un pixel blanc (valeur "0"), noir (valeur "1") ou rouge (valeur "2") comme ci-contre :



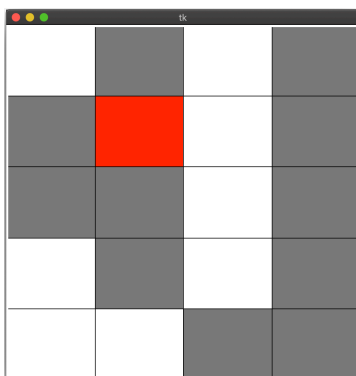
```
dessin = [
    "01111110",
    "01000001",
    "10000010",
    "10001111",
    "01100010",
    "00011100",
]
```

On dispose de la fonction récursive colorier(T, pos_i, pos_j) ci-dessous.

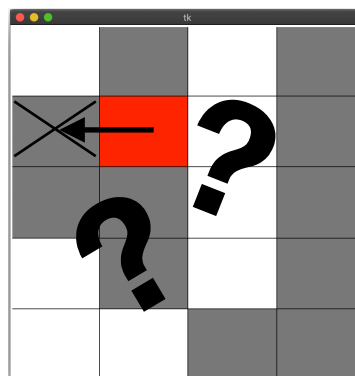
Cette fonction colorie une image représentée par le tableau T ci-dessus à partir du pixel de coordonnées [pos_i][pos_j]. On rappelle que le numéro de la ligne est donné par pos_i et le numéro de la colonne par pos_j.

```
def colorier(t, pos_i, pos_j):
    if t[pos_i][pos_j]=="1":
        return
    else:
        t[pos_i][pos_j]="2"
        affiche(t)
        colorier(t, pos_i+1, pos_j)
        colorier(t, pos_i-1, pos_j)
        colorier(t, pos_i, pos_j+1)
        colorier(t, pos_i, pos_j-1)
```

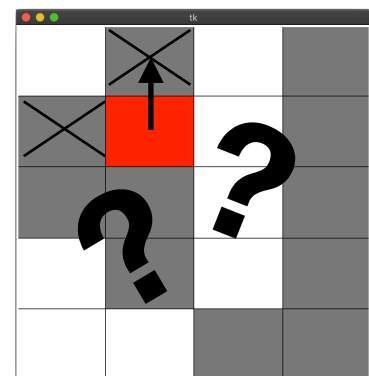
a. Exécutez la fonction colorier à partir de l'étape 0 ci-dessous. On représentera graphiquement l'état du dessin : **chaque étape** de l'algorithme sera indiquée à l'aide d'une flèche comme indiquée ci-dessous.



Étape 0



Étape 1



Étape 2

b. En déduire l'erreur commise dans cette récursion. Proposez une correction.