CORRECTION SUJET NSI n°08

1. EXERCICE 1

Le programme permet de rechercher un élément dans une liste, on crée une variable `taille_tab` qui parcourt le tableau de nombre entiers, on crée une variable `indice` de valeur 0

```
def recherche(elt,tab):  
    taille_tab = len(tab)  
    indice = 0
```

le programme nous dis que quand l'indice est inférieur à la taille du tableau et et que l'indice du tableau est différent de l'élément, on ajoute +1 à l'indice

```
while indice < taille_tab and tab[indice] != elt:  
    indice += 1
```

le programme nous indique si l'indice est égal à la taille du tableau alors on renvoie -1 sinon on renvoie l'indice

```
if indice == taille_tab:  
    return -1  
else:  
    return indice
```

2. EXERCICE 2

```
def insere(a, tab):  
    l = list(tab) #l contient les mêmes éléments que tab  
    l.append(a)  
    i = len(l) - 2  
    while a < l[i] and i >= 0:  
        l[i+1] = l[i]  
        l[i] = a  
        i = i - 1  
    return l
```

Le programme étant déjà construit, le terminer peut paraître simple, mais il faut faire attention à ne pas tomber dans un piège. Il faut donc procéder par élimination afin d'éviter tout bug ou erreur au sein du bug.

a. $i = \dots$ (ligne 4)

i est nécessaire dans la fonction suivante, le while, et sera réutilisé à l'intérieur du while, de ce fait, cela doit être une valeur de type int. L'une des valeurs qui est de type int, c'est la taille de tab. Ce que je peux donc faire, c'est compléter la ligne suivante.

→ $i = \text{len}(l) - 2$

Pourquoi 2 ? Pour la simple raison que l'on insère un nombre au sein de la liste, donc -1, et que l contient le chiffre a, donc -2.

b. $i = \dots$ (ligne 8)

Le second i est simple à compléter, du fait qu'il soit nécessaire pour le while (while $a < \dots$ and $i \geq 0$), i est simplement une valeur à diminuer.

→ $i -= 1$ ou $i = i - 1$

c. while (ligne 5) + lignes 6 et 7

Pour le while, il faut compléter la partie du “ $a < \dots$ ”, sachant que a est plus grand que la valeur, la valeur est donc la suivante.

→ while $a < l[i]$ and $i \geq 0$:

Enfin, pour les $l[i+1]$ et $l[i]$, il suffit de les attribuer de la manière suivante.

→ $l[i+1] = l[i]$ (modifie le chiffre par celui d'avant)

→ $l[i] = a$ (modifie le chiffre par la valeur que l'on souhaite placer)