Sujet 18

Exercice 2 :

```
1 def est_un_ordre(tab):
2     '''
3     Renvoie True si tab est de longueur n et contient tous les entiers
4     de 1 à n, False sinon
5     '''
6     for i in range(1,len(tab)+1):
7         if i not in tab:
8             return False
9     return True
10
11
12 def nombre_points_rupture(ordre):
13     '''
14     Renvoie le nombre de point de rupture de ordre qui représente un ordre
15     de gènes de chromosome
16     '''
17     assert est_un_ordre(ordre) == True, "ordre n'est pas un ordre de gènes"
18     n = len(ordre)
19     nb = 0
20     if ordre[0] != 1: # le premier n'est pas 1
21         nb = nb + 1
22     i = 0
23     while i < n-1 :
24         if ordre[i]-ordre[i+1] not in [-1, 1]: # l'écart n'est pas 1
25             nb = nb + 1
26             i = i + 1
27     if ordre[-1] != n: # le dernier n'est pas n
28         nb = nb + 1
29     return nb
```

Ligne 6 :

le nombre i doit prendre toutes les valeurs entre 1 et n, n étant la longueur de la liste (donc len(tab))
Il faut penser à rajouter 1 à len(tab) sinon on ne prend pas la valeur n

Ligne 7 :

Dès que l'élément regardé n'est pas dans le tableau on renvoie Faux.
Donc si i n'est pas dans tab.

Ligne 17 :

ordre est un ordre de gènes si la fonction est_un_ordre renvoie True

Ligne 20 :

le premier élément du tableau ordre est ordre[0]

Ligne 24 :

L'écart entre deux éléments correspond au résultat d'une soustraction.
Deux éléments qui se suivent sont l'élément à l'indice i et à l'indice i+1

Ligne 27 :

le dernier élément du tableau ordre est ordre[-1] ou ordre[len(ordre)-1]