

Longueur d'une liste

Question 1:

à quoi correspondent chaque valeur affichée? Compléter le tableau.

instruction	resultat	description
<code>len(liste_chaine)</code>		
<code>liste_chaine[0]</code>		
<code>liste_chaine[0]</code>		

```
i = 0
print(liste_chaine[i])
i = i + 1
print(liste_chaine[i])
```

Question 2: Que valent `i` et `liste_chaine[i]` dans chacun des cas?

variable	valeur
<code>i</code> au debut	
<code>liste_chaine[i]</code>	
<code>i</code> après <code>i = i + 1</code>	
<code>liste_chaine[i]</code>	

Question 3: Compléter: dans l'instruction `for i in range(n):`, le variant de boucle, c'est la variable Ce variant aura pour valeurs successives: ..., ..., ..., ...

Question 4: Afficher les éléments de la liste `liste_chaine` à l'aide d'une boucle bornée. Le programme doit afficher:

```
lapin
chat
chien
chiot
dragon
ornithorynque
```

```
liste_chaine = ["lapin", "chat", "chien", "chiot",
                "dragon", "ornithorynque"]
n = len( # a completer
for i in range(n):
```

Question 5: Afficher les éléments de la liste `liste_chaine` ainsi que la longueur de chaque mot. Utiliser une boucle bornée.

Le programme doit afficher:

```
lapin 5
chat 4
chien 5
chiot 5
dragon 6
ornithorynque 13
```

```
n = len(
```

Division entiere et slice

Question 1:

- En partant de la fin: Reperer une serie d'octets identiques Combien de bits constituent l'en-tête?

```
message_code = '011011010001100010011000100110001001100010'
```

- Ecrire une instruction qui élimine les bits de l'en-tête avec un slice. Stocker dans des variables différentes l'en-tête et le texte.
- afficher texte

```
en-tete =
texte =
```

Question 2: Calculer avec une division entiere de `len(texte)` et stocker dans n. Afficher n.

Question 3: Compléter le script pour obtenir la liste L1:

```
['bonjour', 'comment', 'allez', 'vous']
```

```
L1 = [] # d'abord creer une liste vide
L1.append('bonjour')
L1.append('comment')
L1.
```

Question 4: Utiliser une boucle bornée pour mettre les entiers successifs dans une liste.

On veut obtenir:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
L = []
n = ...
for i in range(n):
```

Question 5: Découper texte avec un slice pour chaque octet:

```
octet1 = texte[0:8]
octet2 = texte[8:16]
octet3 =
octet4 =
```

Remarquez la suite numérique utilisée pour faire un découpage tous les 8 bits. On peut écrire une formule en fonction d'un variant i:

```
pour i variant de 0 à 3:
    ajouter texte[8 * i: 8 * (i+1)] dans la liste L
```

a l'aide d'une boucle bornée, ajouter chaque octet dans une liste. Utiliser l'algorithme précédent.

Afficher à nouveau L. Vous devriez obtenir

```
['01100010', '01100010', '01100010', '01100010']
```

```
L = []
n = len(texte)//8
for i in range(n):
```

Code Cesar

Question 1: Calculer les valeurs décimales des nombres binaires suivants: utiliser `int(binaire,2)`

```
int('10001100',2)
# affiche 140

'01101110'
'01101111'
'01110000'
```

Question 2: Pour chaque element de la liste L, calculer la valeur entiere correspondante. Stocker la valeur dans une nouvelle liste L2.

Utiliser la fonction `int(b,2)`, où b est une chaine de 8 caracteres binaires.

```
L2 = []
for b in L:
    nombre =
    L2.append(
L2
```

Question 3: déterminer les valeurs des codes ascii, exprimées en entier décimal, pour les caractères suivants. Utilise la fonction : `ord(caractere)`

```
ord('x')
# affiche 120

'a'
'z'
'e'
'r'
't'
'y'
```

Question 4: Pour chaque element de la liste L2, determiner le caractere ascii correspondant. Et le stocker dans une liste L3

```
L3 = []
for i in L2:
    L3.append(
L3
```

Question 5: Utiliser la méthode `join` vue plus haut pour associer les éléments de la liste L3 en un mot de 4 caractères.

Question 6: Reprendre la liste L2 précédente: soustraire 1 à chaque valeur. Stocker les nouvelles valeurs dans L3

```
L3 = []
for i in L2:
    L3.append(
L3
```

Placer les caractères correspondants dans une nouvelle liste L4. Déterminer alors le nouveau mot obtenu. C'est celui du message d'origine, avant chiffrement par décalage d'une unité.

```
L4 = []  
for i in L3:  
    L4.append(  
L4
```

Décryptage

Question 1: quelles sont les différentes étapes utilisées? Quel est le message obtenu?

Question 2: Recopier le script de la fonction.