

NOM et Prénom :

Mercredi 5 février 2020

1^{ère} – Spé N.S.I.

Interrogation n°2 (55 min.)

(Aucun document autorisé – Ordinateur et calculatrice interdite)

Compléter les réponses directement sur le sujet.

Remarque : si l'instruction demandée doit provoquer une erreur, écrire simplement « Erreur ».

Exercice 1 (4 points)

Donner le contenu des variables définies par :

a = 5 + 2	
b = "5" + 2	
c = "5" * 2	
d = 5 ** 2	

e = 5 / 2	
f = 5 // 2	
g = "5" // 2	
h = 5 % 2	

Exercice 2 (4 points)

Sachant que l'on a : liste = [2, 3, 5, 7]

et chaine = "bonsoir"

Donner le contenu des variables définies par :

a = liste[2]	
b = liste[1:3]	
c = liste*2	
d = 2 in liste	

e = chaine[0]	
f = chaine[:3]	
g = chaine[2:]	
h = "a" in chaine	

Exercice 3 (4 points)

Ecrire la fonction *nb_voyelles*(mot) qui compte le nombre de voyelles dans le mot entré en paramètre.

(On considérera que le mot est en minuscules non accentuées)

Définition de la fonction <i>nb_voyelles</i> (mot)	
Initialisation du compteur nb à 0	
Initialisation de la variable voyelles à "aeiouy"	
Pour chaque caractère c du mot	
Si le caractère c est dans la liste des voyelles	
Alors augmenter le compteur nb de 1	
Renvoyer la valeur du compteur nb	

.../...

Exercice 4 (4 points)

Voici la fonction hamming(mot_1, mot_2) qui calcule la « distance de Hamming » entre deux mots :

- Si les deux mots ont la même longueur, la distance est égale au nombre de positions pour lesquelles les deux mots ont des caractères différents.
- Si les deux mots n'ont pas la même longueur, renvoyer la valeur (-1).

Mais il y a 8 lignes où se trouve une erreur (faute de frappe, erreur de syntaxe, ...).

Retrouver les... Entourer l'erreur trouvée à gauche, et corriger de l'erreur à droite.

1. def hamming(mot_1 ; mot_2) :	1.	
2. Distance de Hamming entre deux mots	2.	
3. if len(mot_1) = len(mot_2) :	3.	
4. dist = 0	4.	
5. for i in range(mot_1) :	5.	
6. if mot_1(i) != mot_2(i) :	6.	
7. dist =+ 1	7.	
8. else	8.	
9. dist = -1	9.	
10. return dist	10.	

Exercice 5 (4 points)

Voici une fonction calculant ce que l'on appelle la durée de vol et l'altitude maximale d'une suite de Syracuse.

Qu'obtient-on dans la console en lançant les instructions suivantes ?

<pre>def syracuse(a, details = False) : """ Suite de Syracuse : Si a est pair, le suivant est a//2 Si a est impair, le suivant est 3*a + 1 """ if details : print("Valeur de départ :", a) compteur = 0 alt_max = a while a != 1 : compteur += 1 if a%2 == 0 : a = a//2 else : a = 3*a + 1 if details : print(compteur, ":", a) if a > alt_max : alt_max = a return [compteur, alt_max]</pre>	a) print(syracuse(1))
	b) print(syracuse(32))
	c) print(syracuse(5, True))
d) print(syracuse(13))	