Lycée Charlemagne

.2018.– *MPSI2* –.2019.

JEUDI NOVEMBRE

1FO 1

Le professeur a demandé de créer une procédure qui prend en entrée un entier naturel n et retourne la somme de ses diviseurs propres [1]. C'est ainsi que `diviseurs(13)` donnera 1 et `diviseurs(14)` donnera 10 et `diviseurs(54)` donnera 66. L'élève a proposé le script suivant :

```
def diviseurs(n) :
....for k in range(n) :
........if n%k = 0 :
............s += k
....print(s)
```

Rectifiez son script, en expliquant les erreurs commises. 3 pt.

Un nombre est dit

	DÉFICIENT	PARFAIT	ABONDANT
si	`diviseurs(n)<n`	`diviseurs(n)=n`	`diviseurs(n)>n`
exemple	`7, 8, 9`	`6, 28`	`12, 20`

Ecrivez un script qui donne la liste des nombres DÉFICIENTS plus petits que 1000. 3 pt.

On définit

```
def chaine(n) :
....nn = n
....L = [nn]
....while diviseurs(nn) > nn :
........nn = diviseurs(nn)
........L.append(nn)
....return(nn)
```

Donnez le résultat de `chaine(66)`. Donnez le résultat de `chaine(20)`. Donnez le résultat de `chaine(24)`. 6 pt.

Expliquez ce que fait le programme `chaine` [2]. 2 pt.

Ecrivez un programme qui affiche `chaine(n)` pour tous les entiers `n` plus petits que 500 dont la `chaine` est au moins de taille 5. 2 pt.

Un entier est dit ABONDANT PRIMITIF si il est abondant et que tous ses diviseurs propres sont déficients.

Vérifiez à la main que 70 est ABONDANT PRIMITIF. 2 pt.

Ecrivez un script qui prend un entier naturel `n` et indique si il est ABONDANT PRIMITIF. 2 pt.

A toutes fins utiles, pour ne pas vous casser la tête, je vous offre une liste des premiers nombres premiers (plus un intrus que vous saurez facilement détecter...) :

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 175, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281

---

1. un diviseur propre de n est un entier qui divise n, mais pas n lui même
2. non pas en disant comme me le font certains élèves « il prend un entier n, il donne à nn la valeur de n, il crée une liste L dont l'élément est nn, tant que diviseurs(nn) est plus grand... ». on ne demande pas la paraphrase, on veut l'explication en mots du résultat