

CC3 Dakar

2h B.Caillier

Tous documents sont autorisés, en particulier le polycopié de cours
<http://pascal.ortiz.free.fr/contents/python/decouverte/decouverte.pdf>.
Aucune communication entre étudiants
PAS de INPUT. (pénalité sinon!!)
Bien renommer les fichier Exo_1_nom_prenom.py (pénalité sinon!!)
Pas de copier-coller de code (pénalité sinon!!)...

1 Étude d'une suite définie par une somme (3pts)

On pose $n \geq 1$ entiers, $S_n = \sum_{k=1}^n \left(\frac{k}{n}\right)^n$. Par exemple $S_5 = \frac{1}{5} + \frac{4}{25} + \frac{27}{125} + \frac{256}{625} + \frac{3125}{3125} = 1.9856$

1. En calculant , les 20 premiers termes de la suite, et en stockant ces valeurs dans une liste, déterminer si la suite S_n est monotone (autrement dit, elle croît toujours ou elle décroît toujours).
2. La suite semble-t-elle tendre vers une certaine valeur ?

2 Divisions successives par deux (3pts)

On part d'un entier $n > 1$ et s'il est pair, on le divise par 2 et s'il est impair, on lui ajoute 1 et on divise par 2. Si par exemple $n=42$ on obtient 21 et si $n=99$ on obtient 50. Maintenant, on répète ce procédé jusqu'à ce que le nombre obtenu vaille 1.

1. On demande d'écrire une fonction `divisi(n)` qui prend en paramètre un nombre n et qui renvoie le nombre de divisions effectuées, tout en affichant les étapes successives.

Par exemple, si $n=25$, les différentes étapes affichées seront : 13 7 4 2 1 et le nombre 5 (divisions effectuées) sera renvoyé.

3 Diviseurs, nombre premier (3pts)

Dans cet exercice, tous les entiers considérés sont strictement positifs. On rappelle qu'un entier d est dit un diviseur d'un entier n si n est un multiple de d . Par exemple, 10 est un diviseur de 2020 mais n'est pas un diviseur de 2024.

1. Écrire une fonction `diviseurs(n)` qui renvoie la liste de tous les diviseurs de l'entier n . Voici quelques exemples du comportement de la fonction `diviseurs` :

```
42 -> [1,2,3,6,7,14,21,42]
41 -> [1,41]
81 -> [1,3,9,27,81]
[1] -> [1]
```

2. Un entier est dit premier s'il admet exactement DEUX diviseurs. Par exemple, 42 n'est pas premier car il admet au moins trois diviseurs, par exemple 1, 6 et 42. En revanche, 41 est premier car ses seuls diviseurs sont 1 et 41. Écrire une fonction `estPremier(n)` qui renvoie `True` si l'entier `n` est premier et `False` sinon. La fonction doit utiliser la fonction `diviseur` de la question précédente.

4 Minimum des entiers pairs (4pts)

On donne une liste `L` d'entiers et un indice valide de la liste (l'indice est appelé `debut`) tel que `L[debut]` soit pair.

1. Construire une fonction `mini(L, debut)` qui renvoie la valeur du plus petit des éléments pairs de la liste `L` et ayant un indice supérieur ou égal à `debut`.

Par exemple, si `L=[81, 32, 10, 21, 42, 11, 12, 9, 12, 65, 46]` et si `debut=4` alors `mini(L, debut)` vaut 12.

2. On donne une liste `L` d'entiers contenant au moins un entier pair et on demande d'écrire une fonction `ind_elt_pair(L)` qui renvoie le plus petit indice `i` de `L` tel que `L[i]` soit pair. Il est attendu que votre code utilise une boucle `while`

Par exemple, si `L=[81, 19, 31, 21, 42, 11, 12]` alors `ind_elt_pair(L)` vaudra 4. .

3. On donne une liste `L` d'entiers contenant au moins un entier pair. Écrire une fonction `miniPairs` qui renvoie le plus petit des éléments de `L` qui soit pair.

Par exemple :

```
[81, 32, 12, 9, 12, 65, 46] -> 12
[81, 65, 46] -> 46.
```

Il est attendu que la fonction `miniPairs` utilise les deux fonctions des questions précédentes.

5 Toujours plus (3pts)

On appelle Toujours Plus le jeu utilisant un seul dé à 6 faces et dont la règle est la suivante : on répète le lancer du dé jusqu'à ce que la valeur lue sur le dé diminue strictement par rapport au lancer précédent. Autrement dit, si le lancer de dé est supérieur ou égal au lancer précédent, la partie continue, sinon, la partie est terminée. Voici trois exemples de parties complètes de Toujours Plus :

2 4 5 5 2

4 1

5 5 5 6 6 5

En particulier, une partie de Toujours Plus dure au moins 2 coups.

1. Écrire une fonction `jouerToujoursPlus`, sans paramètre et qui simule une partie de Toujours Plus. La fonction devra renvoyer la liste des lancers successifs de la partie.
2. Simuler une partie 100000 fois et calculer la moyenne du nombre de lancer.

6 Critère de divisibilité par 7 (4pts)

Pour savoir si un entier n , écrit en base 10, est multiple de 7, comme c'est le cas de $77 = 7 \times 11$, $42 = 7 \times 6$ ou $490 = 7 \times 70$, on peut utiliser le procédé suivant :

- on supprime le chiffre des unités u de n ce qui donne un entier nommé m
- on calcule $N = m + 5u$
- Alors n est multiple de 7 si et seulement N l'est.

Si nécessaire, on peut recommencer le procédé avec N et cela jusqu'à ce qu'on rencontre un entier assez petit que l'on sait être (ou ne pas être) multiple de 7.

Par exemple, 69132 est-il un multiple de 7 ? On applique la règle plusieurs fois de suite :

- $6913 + 5 \times 2 = 6923$
- $692 + 5 \times 3 = 707$
- $70 + 5 \times 7 = 105$
- $10 + 5 \times 7 = 35$
- $3 + 5 \times 5 = 28$
- $2 + 5 \times 8 = 42$
- $4 + 5 \times 2 = 14$
- $1 + 5 \times 4 = 21$
- $2 + 5 \times 1 = 7$ jusqu'à obtenir 7.

1. Écrire une fonction récursive `est_multiple_7(n)` qui partant d'un entier n applique l'algorithme ci-dessus et renvoie `True` si n est multiple de 7 et `False` sinon.