

CC2 : Nombres premiers

Durée de l'épreuve : 1h50

Vous pouvez répondre aux questions de manière concise, mais la précision de vos réponses sera un élément important de la notation des copies. À elles seules, les parties I et II peuvent donner une note fort respectable (17/20 si les réponses sont correctes et brièvement justifiées) et la partie III permettra de gagner des points supplémentaires (et/ou de compenser d'éventuelles erreurs dans les parties précédentes).

On rappelle que pour $n \in \mathbb{N}$ et $m \in \mathbb{N}^*$, l'instruction $n\%m$ calcule le reste de la division (euclidienne) de n par m .

On rappelle aussi que pour deux entiers $a < b$, $\text{range}(a, b)$ calcule la liste $[a, a+1, a+2, \dots, b-1]$ (liste qui compte $b - a$ éléments). En revanche si $a \geq b$, alors $\text{range}(a, b)$ est la liste vide (c'est à dire $[]$).

On rappelle enfin que multiplier une liste par un entier la concatène plusieurs fois avec elle-même. Par exemple $[1, 2] * 3$ renvoie $[1, 2, 1, 2, 1, 2]$.

I. Rappels

Les diviseurs positifs d'un entier naturel $n \in \mathbb{N}$ sont les entiers $d \in \mathbb{N}$ pour lesquels il existe $q \in \mathbb{N}$ tel que $n = dq$.

On dit qu'un entier $p \in \mathbb{N}$ est premier s'il a exactement deux diviseurs positifs.

Les multiples d'un nombre $n \in \mathbb{Z}$ sont tous les nombres $m \in \mathbb{Z}$ tels qu'il existe $q \in \mathbb{Z}$ tel que $m = nq$.

En particulier tout entier est multiple de lui même.

I.1. Exemple

Question 1

On pose $n = 0$. Déterminer l'ensemble des diviseurs positifs de n , et déduire si n est premier ou pas.

Répétez ensuite cette question pour $n = 1$, puis pour $n = 2$, pour $n = 3$, pour $n = 4$, et pour $n = 5$.

I.2. Crible d'Ératosthène

~~0~~ ~~1~~ (2) (3) ~~4~~ (5) ~~6~~ ...

Le "crible d'Ératosthène" est une méthode permettant de lister tous les nombres premiers plus petits qu'une certaine valeur n :

- On liste tous les entiers inférieurs à n (en commençant par 0)
- Au sein de cette liste, on raye les entiers 0 et 1.
- Deux est alors le plus petit entier à ne pas être rayé. Il est premier (on l'entoure pour s'en souvenir), et on raye tous ses multiples sauf lui-même.
- Le plus petit nombre à n'être ni rayé ni entouré est désormais trois. Il est premier (donc on l'entoure) et on barre tous ses multiples sauf lui même.
- On reproduit le procédé : à chaque étape, le plus petit nombre ni rayé ni entouré est automatiquement un nombre premier (s'il n'est pas rayé, c'est qu'il n'est pas multiple des autres

nombres premiers). On l'entoure et on raye alors tous ses multiples sauf lui même (en tant que multiples ils ne sont pas premiers).

- Une fois que chaque nombre est soit rayé soit entouré, les nombres entourés forment la liste des nombre entiers plus petits que n .

Question 2

À l'aide de cette méthode, déterminer tous les nombres premiers plus petits que 20.

II : Liste des plus petits nombres premiers

On demande à trois personnes d'écrire une fonction `petits_preiers` telle que, quand $n \in \mathbb{N}$, `petit_preiers(n)` détermine la liste des nombres premiers plus petit que n . On ne se préoccupe pas de savoir ce que fait `petit_preiers(n)` si $n \notin \mathbb{N}$.

II.1 Réponse d'une première personne

```
def nb_divs(n):
    nb=0
    for d in range(2,n+1):
        if n%d==0:
            nb=nb+1
    return nb
def est_premier(p):
    return nb_divs(p)==2
def petits_preiers(n):
    r=[]
    for i in range(n):
        if est_premier(i):
            r=r+[i]
    return r
```

Question 3

- Que renvoie python quand on demande `nb_divs(0)` ? de même pour `nb_divs(1)`, `nb_divs(2)`, `nb_divs(3)`, `nb_divs(4)` et `nb_divs(5)` ?
- En déduire ce que renvoie python quand on demande `petits_preiers(6)`.
- Ce programme produit-il bien la liste des nombres premiers plus petits que n , et dans le cas contraire comment pensez-vous qu'il faudrait le corriger ?

II.2 Réponse d'une deuxième personne

```
def est_premier(p):
    for i in range(2,p):
        if p%i==0: return False
    return True
def petits_preiers(n):
    return [p for p in range(n) if est_premier(p)]
```

Question 4

- Déterminer ce que renvoie Python quand on calcule "est_premier(0)" avec la fonction définie ci-dessus. Déterminer de même ce que renvoient est_premier(1), est_premier(2), est_premier(3), est_premier(4), et est_premier(5).
- En déduire, pour le programme ci-dessus, ce que renvoie petits_premiers(6) ?
- Ce programme produit-il bien la liste des nombres premiers plus petits que n, et dans le cas contraire comment proposeriez-vous de le corriger ?

II.3 Réponse (incomplète) d'une troisième personne

Une troisième personne a répondu en s'appuyant quant à elle sur le crible d'Ératosthène, mais il manque la fin de son programme :

```
def marque_multiples(L,p):
    k=2
    while (p*k)<len(L):
        L[p*k]="rayé"
        k=k+1

def petits_premiers(n):
    L=["vierge"]*n
    L[0]="rayé";L[1]="rayé"
    for p in range(n):
        if L[p]=="vierge":
            L[p]="encerclé"
            marque_multiples(L,p)
    return [p for p in range(n) if
```

Question 5

- Quand on exécute petits_premiers(6), que vaut la liste L avant la boucle "for p in range(n)" ?
- De même, après chaque itération de la boucle, indiquer ce que vaut la liste L.
- Compléter la fin du programme pour que petits_premiers(n) renvoie effectivement la liste des nombres premiers plus petits que n.

III. Décomposition en facteurs premiers

Question 6

Écrire une fonction fac_prem qui factorise un entier positif en produit de nombres premiers. Par exemple, fac_prem(20) devra renvoyer [2, 2, 5], car $20=2\times 2\times 5$.