

Exercices - Concours Commun INP - Filière MP épreuve 2

- Corrigé (très) partiel : corrigé

AVERTISSEMENT : Ceci n'est pas une correction *in extenso* de l'épreuve. Il s'agit plutôt d'une lecture personnelle des questions, avec des indications, des idées de preuve, des mises en garde d'erreurs à éviter. Ce n'est surtout pas une correction modèle à reproduire... Pour signaler toute erreur, merci d'écrire à devgeolabo@gmail.com

Exercice 1

1. On traite le cas $n = 1$ séparément, puis on utilise l'algorithme proposé par l'énoncé. On teste tous les nombres entre 2 et \sqrt{n} pour voir s'ils divisent n . Si c'est le cas pour un entier, on retourne False. Sinon, on retourne True.

```
def estPremier(n):
    if (n<2):
        return False
    d=2
    while ((d*d)<=n):
        if ( (n%d)==0):
            return False
        d+=1
    return True
```

2. On parcourt tous les nombres compris entre 2 et n , et on les ajoute à la liste s'ils sont premiers.

```
def liste_premiers(n):
    l=[]
    for d in range(2,n+1):
        if (estPremier(d)):
            l.append(d)
    return l
```

3. On initialise un entier s à 0. On augmente s de 1 à chaque fois que p divise n , et on divise n par p .

```
def valuation_p_adique(n,p):
    s=0
    while ((n%p)==0):
        s+=1
        n=n/p
    return s
```

4. Avec la même idée :

```
def val(n,p):
    if ((n%p)==0):
        return val(n/p,p)+1
    else:
        return 0
```

Exercices - Concours Commun INP - Filière MP épreuve 2

- Corrigé (très) partiel : corrigé

5. On parcourt la liste des premiers inférieurs ou égaux à n , et on ajoute ce premier p et la valuation p -adique de n à la liste si toutefois cette valuation est strictement positive.

```
def decomposition_facteurs_premiers(n):
    l=liste_premiers(n)
    liste=[]
    for p in l:
        vp=val(n,p)
        if vp>0:
            liste.append( [p,vp] )
    return liste
```
