TP d'informatique n°7

" Boucles, structures conditionnelles, calcul de complexité :
Autour des nombres premiers"

But du TP
- Ecrire les premières structures conditionnelles et les premières instructions itératives
- Consolider les connaissances sur les structures conditionnelles et itératives.
- Découvrir la syntaxe des fonctions

Vous penserez **à sauvegarder régulièrement** votre travail dans un fichier dans votre espace personnel. Pour le nom de fichier, n'utilisez ni accent, ni espace, ni caractères spéciaux (autres que – et _) : en effet, certains projets nécessitent l'ouverture de fichiers que l'on aura écrits au préalable : or cet appel peut s'avérer délicat si le nom de ce fichier possède des caractères "non habituels"

A. Autour des nombres premiers

1. Ecrire des séquences d'instructions qui, étant donné un entier n naturel (qu'on supposera supérieur ou égal à 4), retourne le booléen True si n est un nombre premier et False sinon. Tester sur n = 4523, 1234567 ou 1234567111111.

On utilisera l'algorithme suivant :

res ← True
pour k allant de 2 à √n **faire** Rem : il faudra peut – être aller jusqu'à 1 + E(√n)
    si k divise n **alors**
        res ← False
Résultat : res

2. Charger le fichier texte appelé **est_premier** dans la page de cours en ligne du site internet du lycée. **R**ecopier le contenu de ce fichier dans votre session de travail. **E**xécuter la commande suivante (après l'avoir écrite dans l'éditeur) :

```
for n in range (1, 50) :
    if est_premier(n) :
        print(n)
```

3. Combien y-a-t-il d'entiers naturels inférieurs à 100 qui sont premiers ? Même chose avec 10^k pour k dans {3, 4, 5, 6}.

On utilisera l'algorithme suivant :

cpt ← 0
pour n allant de 1 à N **faire** N est la borne demandée
    si n est premier **alors**
        cpt ← cpt + 1
Résultat : cpt

4. Déterminer les $100^{\text{ième}}$ et $1000^{\text{ième}}$ nombres premiers.

5. Combien existe-il de $n < 10^6$ tels que n et n+2 sont premiers ?

6. Trouver le plus petit entier n supérieur à 10^{10} tel que n et n+2 sont premiers.

7. Lire le code de la fonction **est_premier** : combien réalise-t-elle d'opérations élémentaires pour tester la primalité d'un entier n dans le pire des cas ? dans le meilleur des cas ?
Combien y-a-t-il d'opérations élémentaires pour déterminer ma primalité de tous les entiers < N ? (on donne le résultat suivant : la proportion d'entiers < N qui sont des nombres premiers est de l'ordre de grandeur 1/ln(N))

TP d'informatique n°7 (Python Maths)

A. Autour des nombres premiers

1. Ecrire des séquences d'instructions qui, étant donné un entier n naturel (qu'on supposera supérieur ou égal à 4), retourne le booléen True si n est un nombre premier et False sinon. Tester sur n = 4523, 1234567 ou 1234567111111.

```
res = True
for k in range(2, 2 + int(n**0.5)) :
        if n % k == 0:
            res = False
print(res)
```

4523 est premier
1234567 n'est pas premier (il est divisible par 127)
1234567111111 n'est pas premier (divisible par 2243)

2. Charger le fichier texte appelé est_premier dans la page de cours en ligne du site internet du lycée. **R**ecopier le contenu de ce fichier dans votre session de travail. **E**xécuter la commande suivante (après l'avoir écrite dans l'éditeur) :

for k in range (1, 50) : **if** est_premier(n) : print(n)

3. Combien y-a-t-il d'entiers naturels inférieurs à 100 qui sont premiers ? Même chose avec 10^k pour k dans {3, 4, 5, 6}.

```
cpt = 0
for n in range(2, N) :
        if est_premier(n) :
            cpt = cpt + 1
print ( cpt )
```

Il y a 25 nombres premiers inférieurs à 100
Il y a 168 nombres premiers inférieurs à 10^3
Il y a 1229 nombres premiers inférieurs à 10^4
Il y a 9592 nombres premiers inférieurs à 10^5
Il y a 78498 nombres premiers inférieurs à 10^6

4. Déterminer les 100$^{\text{ième}}$ et 1000$^{\text{ième}}$ nombres premiers.
Le 100$^{\text{ième}}$ nombre premier est 541 , le 1000$^{\text{ième}}$ est 7919

5. Combien existe-il de n <10^6 tels que n et n+2 sont premiers ?
Il y a 8169 nombres premiers jumeaux inférieurs à 10^6

6. Trouver le plus petit entier n supérieur à 10^{10} tel que n et n+2 sont premiers.
Le plus petit n tel que n et n+2 soient premiers est : 10 000 000 277

7. Lire le code de la fonction est_premier : combien réalise-t-elle d'opérations élémentaires pour tester la primalité d'un entier n dans le pire des cas ? dans le meilleur des cas ?
Lorsque n est premier, on effectue au maximum $\sqrt{n}$ opérations.
Lorsque n est pair, on en effectue 1
Lorsque n est composé impair, on en effectue entre 1 et $\sqrt{n}$
Combien y-a-t-il d'opérations élémentaires pour déterminer ma primalité de tous les entiers < N ? (on donne le résultat suivant : la proportion d'entiers < N qui sont des nombres premiers est de l'ordre de grandeur 1/ln(N))
Le coût pour les nombres premiers est de $\sqrt{n}$ * n / ln(n)
Le coût pour les composés est plus délicat à déterminer… mais il y a alors de fortes probabilités pour que ce nombre possède un "petit facteur"….

	N = 10^2	N = 10^3	N = 10^4	N = 10^5	N = 10^6
Nombre de nombres premiers	25	168	1 229	9 592	78 498
Nombre total de divisions	236	5 288	117 527	2 745 694	67 740 404
Nombre total de divisions pour les nombres premiers	113	3 057	76 242	1 933 897	50 658 939