

### Exercice :

- Le nombre premier est tout entier naturel supérieur à 1 qui possède exactement 2 diviseurs lui-même et 1.
  - On appelle diviseur propre de  $n$ , un diviseur quelconque de  $n$  avec  $n$  exclu.
  - Un entier naturel est dit parfait s'il est égal à la  $\Sigma$  de tous ses diviseurs propres.
  - Les nombres  $a$  tq  $(a+n+n^2)$  est premier  $\forall n$ , où :  $0 \leq n \leq a-1$  sont appelés nombres chanceux.
- La f<sup>ct</sup> "SomDiv" retourne la  $\Sigma$  des diviseurs propres de son argument.
  - La f<sup>ct</sup> "estParfait" vérifie si son argument est parfait, elle retourne un booléen.
  - La f<sup>ct</sup> "estPremier" vérifie si son argument est premier, elle retourne un booléen.
  - La f<sup>ct</sup> "estChanceux" vérifie si son argument est chanceux, elle retourne un booléen.
  - Ecrire un programme principal (test-Parfait-chanceux.py) qui doit comporter une initialisation de 2 listes parfaits et chanceux, c'est une boucle de parcours de l'intervalle  $[2, 1000]$  incluant les tests des 4 f<sup>ct</sup> nécessaires pour remplir ces listes.

### Réponse :

```
def SomDiv(n):  
    S = 0  
    for i in range(1, n//2 + 1):  
        if n%i == 0:  
            S += i  
    return (S)
```

```
def estParfait(n):  
    return (SomDiv(n) == n)
```

```
def estPremier(n):  
    if SomDiv(n) == 1:  
        return True  
    else:  
        return False
```

```

def estChanceux(a):
    for n in range(0, a-1):
        if (estPremier(a+n+n**2)):
            return True
    else:
        return False

```

```

C = []
P = []
for i in range[2, 1000]:
    if estParfait(i):
        P.append(i)
    if estChanceux(i):
        C.append(i)

```

### Exercice:

Écrire une fct qui échange les clés et les valeurs d'un dictionnaire.

### Réponse:

```

def echange(dict):
    aux = 0
    for dict_value, dict_key in dict.items():
        aux = dict_value
        dict_value = dict_key
        dict_key = aux

```

### ou bien:

```

def echange(dico):
    dico_inv = {}
    for i in dico:
        item = dico[i]

```



`dicc_inv[item] = i`

`return dicc_inv`

### Exercice 1: DS:

1. `L = []`

`for i in range(0, 7):`

`n = int(input("entrer un entier"))`

`if n not in L:`

`L.append(n)`

`print("la liste des nombres saisis est :", L)`

2. `Nombres = [x for x in range(10)]`

3. `Entiers.extend(Nombres)`

4. `Entiers.reverse()`

5. `Liste = [x * 2 for x in Entiers if x % 2 == 1]`

6. `f = open("myFile.txt", "w")`

`f.write("Python est le meilleur langage de programmation")`

`f = open("myFile.txt", "r")`

`contenu = f.read()`

`mots = contenu.split()`

`del mots[3]`

7. `f = open("myFile.txt", "r")`

`text = f.read()`

`mots = text.split()`

`d = {mot, len(mot) for mot in mots}`



2. def partition (dictionnaire):

EtudiantAdmis = {key: value for key, value in dictionnaire if value >= 10}

EtudiantNonAdmis = {key: value for key, value in dictionnaire if value <= 10}

return EtudiantAdmis, EtudiantNonAdmis

Exercice 2:

1. def dupliques (liste):

compteur = {}

duplication = []

for element in list:

if element in compteur:

compteur[element] += 1

else:

compteur[element] = 1

for element, compte in compteur.items():

if compte >= 3:

duplication.append(element)

return duplication

2. def dictionnairetrie (d):

dictrie = {}

for cle, valeur in d.items():

if isinstance(valeur, list):

valeurtrie = sorted(valeur)

dictrie[cle] = valeurtrie

else:

dictrie[cle] = valeur

return dictrie