Z L A N G A GE 7 − FONCTION janvier 2022

7 - Fonctions - Exercices fonctions (correction)



Exercice 1

Définir une fonction test_Pythagore qui prend trois entiers a,b et c en arguments et renvoie un booléen indiquant si $a^2+b^2=c^2$, ou $b^2+c^2=a^2$ ou $c^2+a^2=b^2$.

Tests et exemples d'usage:

```
>>> print(test_pythagore(3, 4, 5))
True
>>> print(test_pythagore(7, 2, 12))
False
>>> print(test_pythagore(3, 5, 4))
True
True
```

True False True True





Exercice 2

Définir une fonction valeur_absolue qui prend un entier en argument et renvoie sa valeur absolue.

Exemples et tests:

```
>>> print(valeur_absolue(5))
5
>>> print(valeur_absolue(-5))
5
>>> print(valeur_absolue(0))
0
>>> print(valeur_absolue(-42))
42
```

```
[2]: def valeur_absolue(n):
    if n >= 0:
        return n

    return -n

print(valeur_absolue(5))
print(valeur_absolue(-5))
print(valeur_absolue(0))
print(valeur_absolue(-42))
```

5

5

Ω

42



Janvier 2022

Ехегсісе 3

Créer une fonction multiples pour qu'elle prenne la limite en argument plutôt que d'utiliser la valeur 999. En déduire une fonction $multiples_3 ou_5 (borne_sup)$ qui renvoie la somme des multiples de 3 ou 5 inférieurs ou égaux à $borne_sup$.

Exemples et tests:

ТРИСРСЕ 🛂

```
>>> print(multiples_3_ou_5(2))
0
>>> print(multiples_3_ou_5(6))
14
>>> print(multiples_3_ou_5(10))
33
>>> print(multiples_3_ou_5(100))
2418
```

```
[4]: def multiples_3_ou_6(borne_sup):

somme = 0

for i in range(borne_sup+1):

return somme

print(multiples_3_ou_6(2))

print(multiples_3_ou_6(10))

print(multiples_3_ou_6(10))
```

```
2418
33
0
```

```
>>> print(nb_jours(1,1,2021,15,1,2021))
15
>>> print(nb_jours(1,1,2021,31,12,2021))
365
>>> print(nb_jours(1,1,2024,31,12,2024))
366
>>> print(nb_jours(1,1,1970,31,12,2021))
18993
```

```
[61]: print(nb_jours(1,1,202,1,15,1,1,2021))
                                                                                                                                      return total - nbjours_avant - nbjours_apres
                                                                                                                                                                                                               səənna sərəinrəb tə sərəimərq #
                                                                                                                                                                                                              səp dour uə sınol sər ıəirrəi #
                                                                                                                                                                             total = total + nb_jours_annee(a)
                                                                                                                                                                                                      for a in range(a_debut, a_fin+1):
                                                                                                                                                                                                                                                                                                                                           total = 0
                                              nil ob osnan'i to tudob ob osnan'i ortao eruoi ob eordmon #
                                                  m 'm 'nils | sours = nbjours = apres + nb_jours = apres =
                                                                                                                                                                                                                     for m in range(m_fin + 1, 13):
nil_[ - (nil_m ,nil_s)siom_sruol_dn + sergs_sruoldn = sergs_sruoldn
                                                                                                                                                                                                                                                                                               O = serge_arwolda
                                                                                                    # fin nécessaires pour compléter la dernière année
                                                                                                                                                                                             sp stub ni sárqn sruot sb srámon #
                                    (m , tudeb_s) siom_sruoj_dn + tnsvs_sruojdn = tnsvs_sruojdn
                                                                                                                                                                                                                                 for m in range (1, m_debut):
                                                                                                                                            \label{eq:continuous} \mbox{$\uparrow$} \ \mbox{-} \m
                                                                                                                                                                                                                                                                                              O = Jnsva_aruoįdn
                                                                                                                                                      * départ pour compléter la première année
                                                                                                                                                                                             sp stab al trava sruoj sb srdmon #
                                                            [81]: def nb_jours(j_debut, m_debut, a_debut, j_fin, m_fin, a_fin):
```

01

```
$6681
998
998
```

((1202,21,16,0791,1,1)srwot_dm)tairq

rint(nb_jours(1,1,202,11,1,202,1,1,202,1)) print(nb_jours(1,1,202,4,1,1,202,4))

ACTIVITÉ 4

Exercice 4

Écrire une fonction max2 (a, b) qui renvoie le plus grand des deux entiers a et b.

Exemples et tests:

```
>>> print(max(0, 0))
0
>>> print(max(1, 2))
2
>>> print(max(-2, -1))
-1
```

```
[5]: def max2(a, b):
    if a >= b:
        return a
        return b

print(max(0,0))
    print(max(1,2))
    print(max(-2,-1))
```

0

2

- 1



Exercice 5

Écrire une fonction max3 (a, b, c) qui utilise la fonction max2 de l'exercice précédent et qui renvoie le plus grand des trois entiers a, b, c.

```
>>> print(nb_jours_mois(1904, 2))
29
>>> print(nb_jours_mois(2021, 9))
30
```

```
[13]: def nb_jours_mois(annee, mois):
    if mois == 2:
        if est_bissextile(annee):
            return 29
        else:
            return 28

    if mois == 4 or mois == 6 or mois == 9 or mois == 11:
        return 31

print(nb_jours_mois(1900, 1))
print(nb_jours_mois(1900, 2))
print(nb_jours_mois(1904, 2))
print(nb_jours_mois(2021, 9))
```

31

28

29

30



Exercice 10

En utilisant les fonctions nb_jours_annnee et nb_jours_mois (qui sont disponibles), écrire une fonction nb_jours(j_debut, m_debut, a_debut, j_fin, m_fin, a_fin) qui renvoie le nombre de jours compris entre deux dates données (incluses).

Exemples et tests :

>>> print(nb_jours_annee(2022))

La fonction $\max 2$ est disponible et vous pouvez l'utiliser sans l'implémenter.

Exemples et tests:

[6]: def max3(a, b, c):
 return max2(a, max2(b, c))
 print(max3(0, 0, 0))
 print(max3(0, 0, 0))
 print(max3(0, 0, 0))

21



Exercice 6

Ecrire une fonction puissance(x, k) qui renvoie x à la puissance k (utilisation de l'opérateur ** interdit! évidemment...). On utilisera une boucle f or pour faire le calcul.

On suppose que $k \ge 0$ et on rappelle que $x^0 = 1$.

```
Exemples et tests:
```

```
365

[11]: def nb_jours_annee(annee):

return 366

print(nb_jours_annee(1996))

print(nb_jours_annee(2024))

print(nb_jours_annee(2024))

print(nb_jours_annee(2025))

print(nb_jours_annee(2025))
```



Exercice 9

365

998

998

Écrire une fonction nb_jours_mois(annee, mois) qui renvoie le nombre de jours dans le mois mois de l'année annee. La fonction est_bissextile de l'exercice précédent est disponible (ce qui est bien pratique).

On suppose que le mois mois mois est un entier compris entre 1 (pour janvier) et 12 (décembre).

Exemples et tests:

((1, 0001)siom_sruo [_dn)tnirq <<<</pre>

```
>>> print(puissance(5, 2))
25
>>> print(puissance(100, 0))
1
>>> print(puissance(2, 10))
1024
```

```
[9]: def puissance(x, k):
    resultat = 1
    for i in range(k):
        resultat = resultat * x

    return resultat

print(puissance(5, 2))
print(puissance(100, 0))
print(puissance(2, 10))
```

2511024

LANGAGE

ACTIVITÉ 7

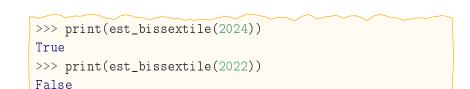
Exercice 7

Écrire une fonction est_bissextile(annee) qui renvoie un booléen indiquant si l'année annee est une année bissextile.

On rappelle qu'une année bissextile est une année multiple de 4 mais pas de 100, ou multiple de 400.

Exemples et tests:

```
>>> print(est_bissextile(1996))
True
```



```
[10]: def est_bissextile(annee):
    est_div_4 = annee %, 4 == 0
    est_div_100 = annee %, 100 == 0
    est_div_400 = annee %, 400 == 0
    if (est_div_4 and not est_div_100) \
        or est_div_400:
        return True

    return False

print(est_bissextile(1996))
print(est_bissextile(2024))
print(est_bissextile(2022))
```

True True

False



Exercice 8

Écrire une fonctions nb_jours_annee (annee) qui renvoie le nombre de jour de l'année annee. La fonction est_bissextile de l'exercice précédent est disponible et vous pouvez l'utiliser (ce qui est bien pratique quand même).

Exemples et tests:

```
>>> print(nb_jours_annee(1996))
366
>>> print(nb_jours_annee(2024))
```