

Notions d'Algorithmes

Premiers pas

EXERCICE 1

- 1) Calculer à la main le résultat pour 2 et $\sqrt{3}$.
- 2) Trouver la fonction f correspondant à cet algorithme où x est le nombre de départ.
- 3) Écrire cet algorithme en Python et le programmer sur votre calculatrice.

```
Choisir un nombre.
Mettre au carré
Multiplier par 10
Ajouter 25
Afficher le résultat
```

EXERCICE 2

On donne la fonction f , écrite en Python

- 1) Tester à la main $f(4)$ et $f(7)$:
- 2) Un élève choisi $n = -3$. Que se passe-t-il? Pourquoi?
- 3) Émettre une conjecture sur l'expression de $f(n)$.
- 4) Démontrer cette conjecture.

```
def f(n):
    q=(n+2)**2
    q=q-(n+4)
    q=q/(n+3)
    return q
```

Tests

EXERCICE 3

Soit f la fonction affine définie par morceaux :
$$\begin{cases} f(x) = -1,5x - 1 & \text{si } x \leq -2 \\ f(x) = 0,25x + 2,5 & \text{si } x > -2 \end{cases}$$

- 1) Calculer, à la main, les valeurs de : $f(-4)$, $f(-2)$ et $f(2)$.
- 2) Écrire un algorithme en Python qui permette de calculer $f(x)$
- 3) Programmer f sur votre calculatrice et tester avec $f(-4)$, $f(-2)$ et $f(2)$.

EXERCICE 4

Écrire un programme en Python qui, à partir des coordonnées de 2 vecteurs $\vec{u}(x; y)$ et $\vec{v}(z; t)$, permet d'afficher le déterminant d et la colinéarité des vecteurs.

On testera cet algorithme avec :

- $\vec{u}(10; -5)$ et $\vec{v}(-4; 2)$
- $\vec{u}(3; -2)$ et $\vec{v}(6; -1)$

EXERCICE 5

Un magasin de reproduction propose les tarifs suivants pour des photocopies

- De 1 à 30 : 0,12 € pièce
- De 31 à 60 : 0,10 € pièce
- Au-delà de 60 : 0,08 € pièce.

- 1) Calculer à la main les prix à payer pour 11, 42 et 80 photocopies
- 2) Montrer que la fonction f associée au prix à payer en fonction du nombre n de photocopies effectuées a pour expressions :
 - Si $n \leq 30$: $f(n) = 0,12n$
 - Si $31 \leq n \leq 60$: $f(n) = 0,1n + 0,6$
 - Si $n > 60$: $f(n) = 0,08n + 1,8$
- 3) Écrire un algorithme en Python donnant le montant à payer en fonction du nombre n de photocopies. On testera cet algorithme avec les résultats de 1)

Boucle conditionnelle

EXERCICE 6

Somme des n premiers naturels

Le programme ci-dessous calcule la somme S_n des n premiers naturels :

$$S_n = 1 + 2 + \dots + n$$

- 1) Rentrer cet algorithme sur votre calculatrice.
- 2) Tester avec les valeurs de n suivantes :

• $n = 6$ • $n = 100$ • $n = 250$ • $n = 1210$

```
def S(n):
    s=0
    for i in range(1,n+1):
        s=s+i
    return s
```

EXERCICE 7

Somme des nombres impairs

- 1) Écrire un programme pour calculer la somme : $S = 1 + 3 + 5 + \dots + (2k + 1)$
 - 2) Remplir le tableau suivant :
- | | | | |
|-----|---|---|----|
| k | 5 | 9 | 19 |
| S | | | |
- 3) Que peut-on faire comme conjecture ?

Synthèse

EXERCICE 8

- 1) On donne la fonction mystere suivante :
« $a \% b$ » signifie reste de a par b et « $! =$ » signifie \neq .
Appliquer à la main cet algorithme avec
 - $a = 391$ et $b = 221$
 - $a = 493$ et $b = 377$
- 2) Écrire ce programme avec votre calculatrice en affichant les valeurs intermédiaires et en le testant avec les valeurs trouvées à la main.
- 3) Remplir le tableau suivant :

```
def mystere(a,b):
    r=a%b
    while r!=0:
        a=b
        b=r
        r=a%b
    return b
```

a	12	18	30
b	8	12	5
Résultat			

Que calcule cet fonction ? Quel est son nom ?

EXERCICE 9**Conjecture de Syracuse**

On considère fonction suivante :

```
def f(n):
    while n>1:
        if n%2==0:
            n=n/2
        else:
            n=3*n+1
    return n
```

- 1) Réaliser, à la main, cet algorithme avec $n = 6$, $n = 7$ et $n = 16$.
- 2) Que constatez-vous?
- 3) Modifier l'algorithme pour qu'il affiche toutes les valeurs de n .
- 4) Modifier l'algorithme pour qu'il affiche le nombre de tests effectués.
- 5) Modifier l'algorithme pour qu'il affiche la valeur maximale de n atteinte.
- 6) Rentrer l'algorithme dans la calculatrice puis remplir le tableau suivant :

n	Nbre d'itérations	Valeur maximale
23		
24		
41		
57		

EXERCICE 10

Compléter l'algorithme suivant pour qu'il affiche la table de multiplication de n (de 0 jusqu'à 12) d'un entier naturel n .

```
def table(n):
    for i in range(...):
        r = ...
        print(...)
```

EXERCICE 11

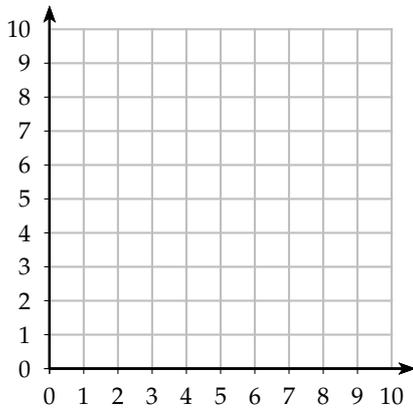
Un distributeur de billets doit donner une somme S avec des billets de 10, 20 ou 50 euros et avec le moins de billets possibles. La somme doit être un multiple de 10 et ne doit pas dépasser 1 000 euros.

- 1) Comment faire pour savoir combien de billets de chaque sorte seront donnés par le distributeur si $S = 330$?
- 2) Écrire une fonction Python qui, en fonction de S , renvoie le nombre de billets de chaque sorte distribués
- 3) Rentrer l'algorithme sur la calculatrice. Le tester avec différentes sommes.



EXERCICE 12

On donne l'algorithme ci-dessous. Appliquer cet algorithme à la main dans le repère ci-dessous. Rentrer le programme sur votre calculatrice pour vérifier



```
import matplotlib.pyplot as plt
for i in range(11):
    plt.plot([0, i], [i, i])
plt.grid(color='0.8')
plt.axis('equal')
plt.show()
```

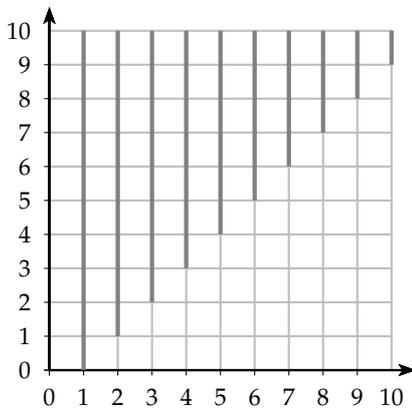
`plt.plot([a, b], [c, d]) :`

Trace le segment des points $(a ; c)$ et $(b ; d)$

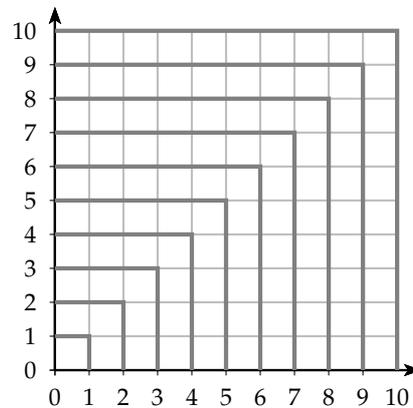
EXERCICE 13**Figures à l'aide d'une boucle itérative**

En vous inspirant du programme précédent, écrire des algorithmes permettant de construire les figures suivantes :

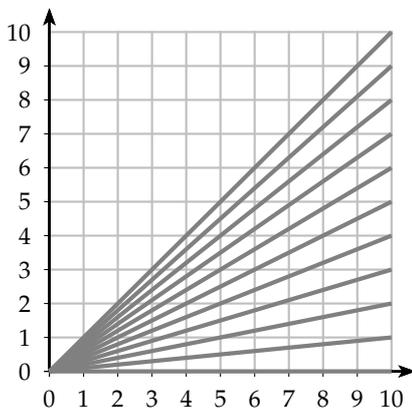
Dessin n° 1



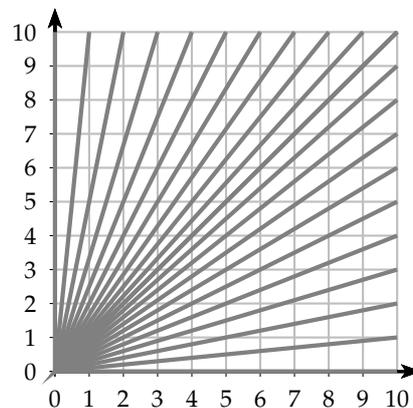
Dessin n° 2



Dessin n° 3

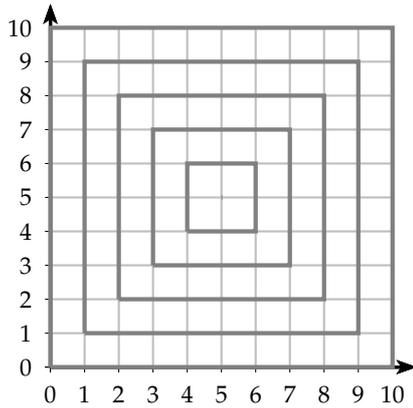


Dessin n° 4

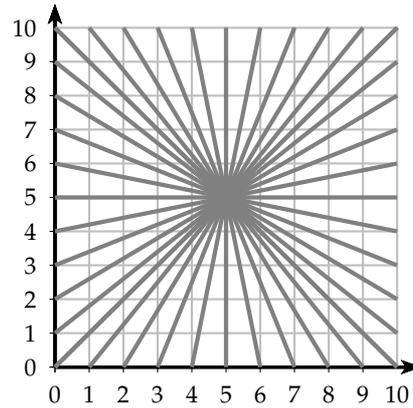


Remarque : Pour le dessin n° 4, penser à une symétrie par rapport à la figure n° 3 (par rapport à la droite d'équation $y = x$)

Dessin n° 5

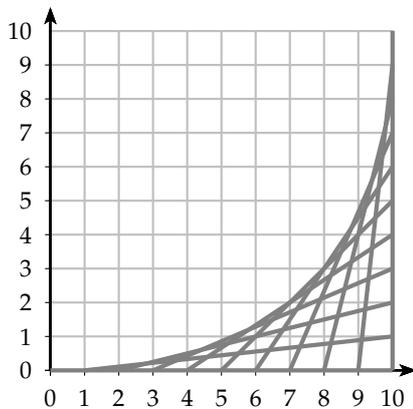


Dessin n° 6

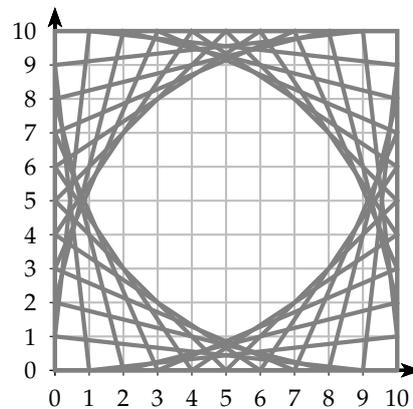


Pour les dessin n° 5 et n° 6, penser à des symétries

Dessin n° 7



Dessin n° 8



Remarque : Pour le dessin n° 8 penser à des symétries du dessin n° 7

EXERCICE 14

Le juste prix

Voici ci-contre un algorithme permettant de jouer "au juste prix" avec un prix entier compris entre 1 € et 100 €

- 1) Rentrer cet algorithme sur votre calculatrice puis testez le.
- 2) Comment peut-on modifier cet algorithme afin de compter le nombre d'essais pour obtenir le juste prix ?

```

from random import*
p=randint(1,100)
n=0
while n!=p:
    n=int(input('donner_un_prix'))
    if n==p:
        print('gagné')
    else:
        if n>p:
            print('inférieur')
        else:
            print('supérieur')

```

EXERCICE 15

On considère le problème suivant :

- On lance une balle d'une hauteur initiale de 300 cm.
- On suppose qu'à chaque rebond, la balle perd 10 % de sa hauteur

On cherche à savoir le nombre de rebonds nécessaires pour que la hauteur de la balle soit inférieure ou égale à 10 cm.

Écrire un algorithme en Python permettant de résoudre ce problème.
Le rentrer sur votre calculatrice et répondre au problème posé.

EXERCICE 16

On voudrait répondre à la question par expérimentation : « Combien faut-il, en moyenne, lancer de fois un dé avant que le 6 soit obtenu pour la première fois ? »

Écrire un programme donnant, à partir de 10 000 expériences aléatoires, une estimation de cette valeur moyenne.