Notions d'algorithme

Premiers pas

EXERCICE 1

1) Appliquer cet algorithme à : 3, -4, 0, $\frac{1}{3}$.
2) Traduire cet algorithme par une fonction f où x est le nombre de départ. Quelle est la nature de cette fonction.
3) Comment choisir le nombre de départ pour que s'affiche le nombre 0 ? le nombre -5 ? Écrire un algorithme traduisant ces calculs.
4) Traduire ce nouvel algorithme par une fonction g où x est le nombre de départ.

Nom : **E1**

Choisir un nombre.
Lui ajouter 1.
Multiplier le résultat par 2.
Soustraire 3 au résultat.
Afficher le résultat.

EXERCICE 2

1) Quel est le résultat pour un nombre de départ de 2 ? $\sqrt{3}$? (faire le calcul à la main)
2) Trouver la fonction f correspondant à cet algorithme où x est le nombre de départ.
3) Écrire cet algorithme en pseudo-code et le programmer sur votre calculatrice.

Nom : **E2**

Choisir un nombre.
Prendre le carré de ce nombre
Le multiplier par 10
Lui ajouter 25
Afficher le résultat

EXERCICE 3

On donne l'algorithme, ci-contre, en pseudo-code

1) Tester, à la main, cet algorithme avec : $N = 4$ et $N = 7$.
2) Un élève a choisi -3. Que se passe-t-il ? Pourquoi ?
3) Émettre une conjecture sur le résultat de cet algorithme.
4) Démontrer cette conjecture.

Nom : **E3**

Variables : N, Q réels
Entrées et initialisation
  Lire N
Traitement
  $(N+2)^2 \rightarrow Q$
  $Q-(N+4) \rightarrow Q$
  $Q/(N+3) \rightarrow Q$
Sorties : Afficher Q

Tests

EXERCICE 4

La valeur absolue d'un réel x, notée $|x|$ est défini par : $|x| = \begin{cases} x & \text{si } x \geqslant 0 \\ -x & \text{sinon} \end{cases}$

On donne l'algorithme ci-contre.

a) Programmer cet algorithme sur votre calculatrice.

b) Tester votre programme avec les valeurs suivantes de x

- $x = 5$ • $x = -4$ • $x = 0$

```
Nom : VA
Variables : X, Y réels
Entrées et initialisation
    Lire X
Traitement
    si X ⩾ 0 alors
        X → Y
    sinon
        −X → Y
    fin
Sorties : Afficher Y
```

EXERCICE 5

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

1) Calculer, à la main, les valeurs de : $f(-4)$, $f(-2)$ et $f(2)$.
2) Écrire un algorithme en pseudo-code qui permette de calculer une image quelconque de la fonction f
3) Programmer la fonction f sur votre calculette et tester avec les images de -4, -2 et 2.

EXERCICE 6

Faire un programme qui, à partir des coordonnées de 2 vecteurs $\vec{u}(x;y)$ et $\vec{v}(z,t)$, qui permette 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 7

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 $n \leqslant 30$: $f(n) = 0,12\,n$
   - Si $31 \leqslant n \leqslant 60$: $f(n) = 0,1\,n + 0,6$
   - Si $n > 60$: $f(n) = 0,08\,n + 1,8$
3) Écrire un algorithme donnant le montant à payer en fonction du nombre n de photocopies. On testera cet algorithme à l'aide des résultats trouvés à la question 1)

Boucle conditionnelle

EXERCICE 8

On appelle partie entière d'un nombre réel x positif ou nul, l'entier noté $\mathrm{E}(x)$ défini par :

$$\text{Si } n \leqslant x < n+1 \quad \text{alors} \quad \mathrm{E}(x) = n$$

On donne le programme ci-contre.

a) Programmer cet algorithme sur votre calculatrice.

b) Tester votre programme avec les valeurs suivantes de x

- $x = 4,347$
- $x = 19,27$
- $x = \sqrt{157}$
- $x = 150$
- $x = 2541,52$

Que constatez vous ? Pourquoi ?

```
Nom : PE
Variables : N entier, X réel
Entrées et initialisation
    Lire X
    0 → N
Traitement
    tant que N + 1 ⩽ X faire
        | N + 1 → N
    fin
Sorties : Afficher N
```

EXERCICE 9

Modifier cet algorithme de façon qu'il puisse calculer la partie entière d'un réel quelconque (positif, négatif ou nul), dont la définition est la suivante :

$$\text{Si pour } n \in \mathbb{Z}, \quad n \leqslant x < n+1 \quad \text{alors} \quad \mathrm{E}(x) = n$$

EXERCICE 10

Somme des N premiers naturels

Le programme ci-dessous calcule la somme S des n premiers naturels, c'est à dire :

$$S = 1 + 2 + \cdots + n$$

a) Programmer cet algorithme sur votre calculatrice.

b) Tester votre programme avec les valeurs suivantes de n

- $n = 6$
- $n = 100$
- $n = 250$
- $n = 1210$

Que constatez vous ? Pourquoi ?

```
Nom : SOMME
Variables : N, I, S entiers
Entrées et initialisation
    Lire N
    0 → S
Traitement
    pour I de 1 à N faire
        | S + I → S
    fin
Sorties : Afficher S
```

EXERCICE 11

Factorielle

Faire un programme pour calculer $n!$ "factorielle n" : $n! = 1 \times 2 \times 3 \times \cdots \times n$

EXERCICE 12

Somme des nombres impairs

a) Faire un programme pour calculer la somme : $S = 1 + 3 + 5 + \cdots + (2k+1)$

b) Remplir le tableau suivant :

k	5	9	19
S			

c) Que peut-on faire comme conjecture ?

Synthèse

EXERCICE 13

Un algorithme célèbre !

1) On donne l'algorithme suivant :
   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 testées à la main.
3) Remplir le tableau suivant :

A	12	18	30
B	8	12	5
Résultat			

Que calcule cet algorithme ? Quel est son nom ?

Nom : **AE**

Variables : A, B, I, R entiers
Entrées et initialisation
  Lire A,B
  $0 \to I$
Traitement
  tant que $\mathrm{E}\left(\dfrac{A}{B}\right) \neq \dfrac{A}{B}$ **faire**
    $A - \mathrm{E}\left(\dfrac{A}{B}\right) \times B \to R$
    $B \to A$
    $R \to B$
  fin
Sorties : Afficher B

* $\mathrm{E}(x)$ signifie la partie entière de x.

EXERCICE 14

Conjecture de Syracuse

On considère l'algorithme suivant :

1) Entrer un entier naturel n.
2) Tant que $n > 1$ réitérer la procédure suivante :
   - Si n est pair remplacer n par $n \div 2$.
   - Sinon remplacer n par $3 \times n + 1$.
3) Afficher la valeur de 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.

Consignes avec la calculatrice

a) Réaliser un programme qui réalise l'algorithme initial (S_0).
b) Tester le programme avec des entiers de votre choix.
c) Modifiez le programme pour qu'il affiche à chaque étape la nouvelle valeur de N et tester à nouveau le programme (S_1).
d) Modifiez le programme pour qu'il affiche le nombre d'itérations et tester à nouveau le programme (S_2).
e) Modifiez le programme pour qu'il affiche la valeur maximale atteinte par N et tester à nouveau le programme (S_3).
f) Remplir le tableau suivant :

n	Nbre d'itérations	Valeur maximale
23		
24		
41		
57		

Exercice 15

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

```
Nom : TM
Variables : N, ..., R entiers
Entrées et initialisation
  | Lire N
Traitement
  | pour ... allant de ... jusque ... faire
  |   |  ... → ...
  |   |  Afficher ...
  | fin
```

Exercice 16

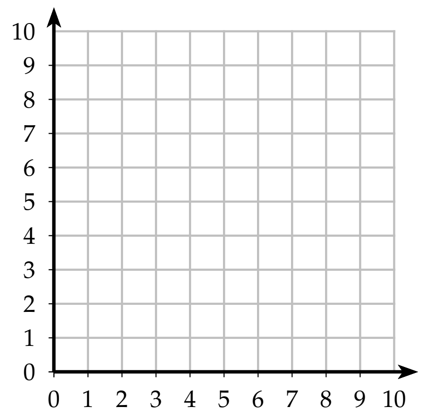
Un distributeur de billets doit donner une somme S avec des billets de 10, 20 ou 50 euros et avec le moins de billets possibles. La somme doit être un multiple de 10 et ne doit pas dépasser 1 000 euros.

1) Comment faire pour savoir combien de billets de chaque sorte seront donnés par le distributeur si $S = 330$?
2) Écrire un algorithme qui demande à l'utilisateur la somme S, lui dit si la somme n'est pas un multiple de 10 ou est supérieure à 1 000 et renvoie le nombre de billets de chaque sorte
3) Écrire le programme sur la calculatrice. Le tester avec différentes sommes.

Exercice 17

On donne l'algorithme ci-dessous.

```
Nom : D1
Variables : K entier
Entrées et initialisation
  | Effacer dessin
Traitement
  | pour K de 1 à 10 faire
  |   | Tracer le segment
  |   | [(0, K), (K, K)]
  | fin
```



Appliquer cet algorithme à la main dans le repère ci-contre. Programmer le ensuite sur votre calculatrice pour vous vérifier

Remarque : Pour effacer le dessin avec la Ti, faire : [dessin] 1 : EffDessin.
Pour tracer un segment avec la Ti faire : [dessin] 2 : Ligne(
Pour programmer le segment [AB] faire : Ligne(x_A, y_A, x_B, y_B)

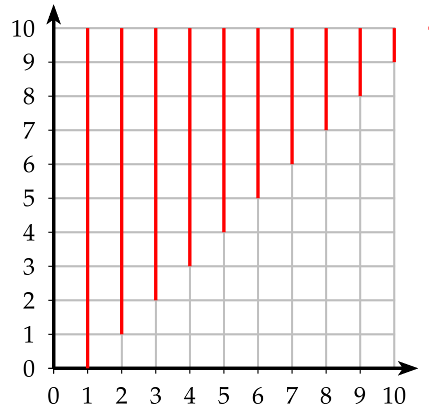
EXERCICE 18

Figures à l'aide de "Pour"

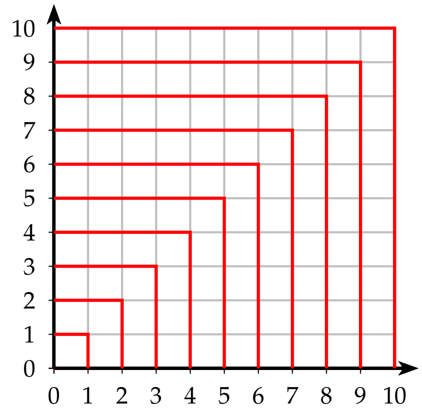
Écrire des algorithmes qui permettent de construire les figures suivantes :

Pour avoir un repère orthonormé, on prendra comme fenêtre : $X \in [0, 15]$ et $Y \in [0, 10]$.

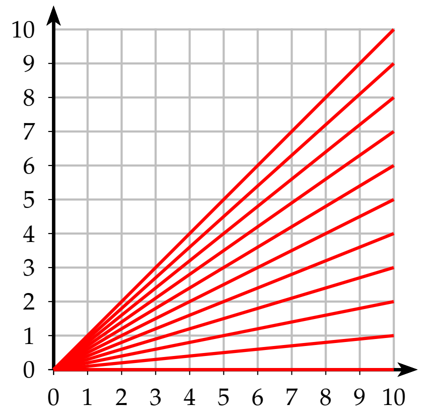
Dessin n° 1



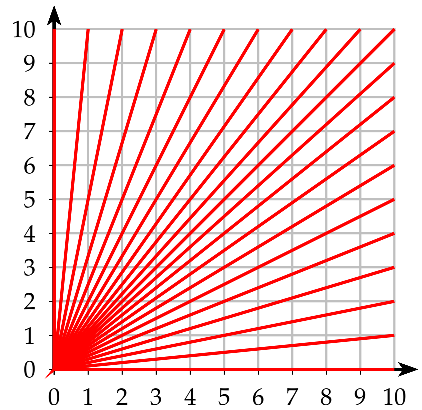
Dessin n° 2



Dessin n° 3

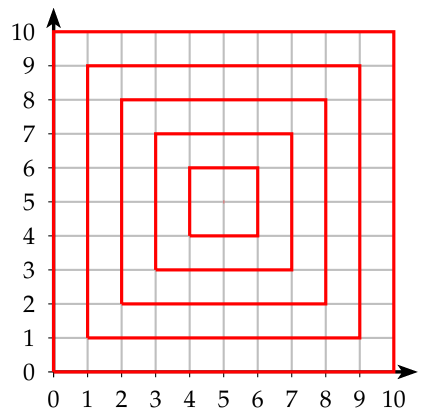


Dessin n° 4

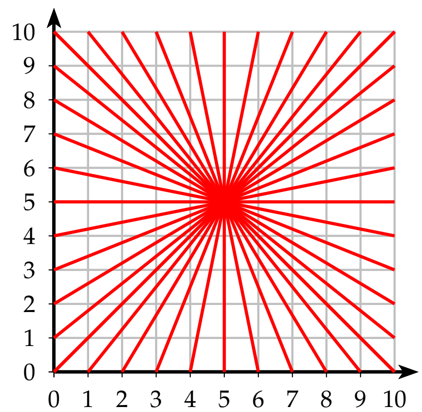


Remarque : Pour le dessin n° 4, on fera une symétrie par rapport à la figure n° 3 (par rapport à la droite d'équation $y = x$)

Dessin n° 5

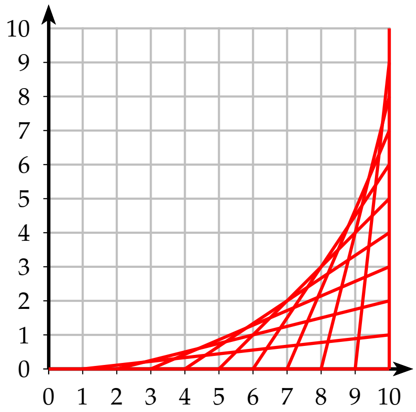


Dessin n° 6

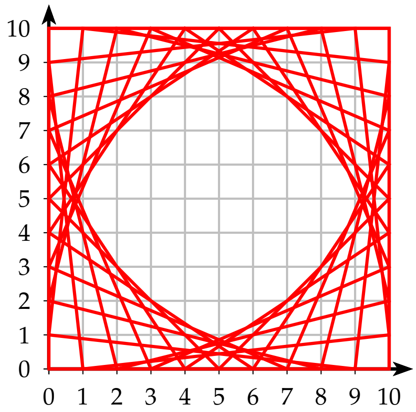


Remarque : Pour les dessin n° 5 et n° 6, penser aux symétries

Dessin n° 7



Dessin n° 8



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

EXERCICE 19

Le juste prix

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

1) Programmer cet algorithme sur votre calculatrice

2) Comment peut-on modifier cet algorithme afin de compter le nombre d'essais afin d'obtenir le juste prix ?

Remarque : Pour entrer un entier aléatoire entre 1 et 100, avec la Ti faire : math PRB 5 : entAléat(1,100)

```
Nom : JP
Variables : N, P entiers
Entrées et initialisation
    entier aléatoire entre 1 et 100 → P
    0 → N
Traitement
    tant que N ≠ P faire
        Lire N
        si N = P alors
            Afficher "GAGNÉ"
        sinon
            si N > P alors
                Afficher "INF"
            sinon
                Afficher "SUP"
            fin
        fin
    fin
```

EXERCICE 20

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 nécessaire pour que la hauteur de la balle soit inférieure ou égale à 10 cm.
Écrire un algorithme permettant de résoudre ce problème.
Le programmer sur votre calculatrice et répondre au problème posé.

EXERCICE 21

Combien faut-il,en moyenne, lancer de fois un dé avant que le 6 soit obtenu pour la première fois ?
Faire un programme donnant, à partir de 10 000 expériences aléatoires, une estimation de cette valeur moyenne.