

1 Algorithme et programme informatique

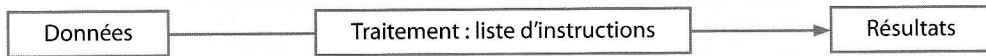
Il y a trois étapes pour écrire un programme informatique :

- analyser le problème posé ;
- écrire un algorithme indépendamment d'un langage de programmation ;
- le traduire dans un langage que comprend la machine que l'on va utiliser : on obtient un programme informatique. On dit qu'on implémente l'algorithme dans un langage de programmation.

Un **algorithme** est une suite d'instructions à suivre qui, à partir de données permettent d'obtenir des résultats clairement définis en un nombre fini d'étapes.

Pour écrire un algorithme on doit envisager trois phases.

- Une **phase préparatoire** : on y repère en particulier les informations à demander à l'utilisateur (nombres, mots, points, listes de nombre etc.) appelées entrées.
- Le **traitement** : c'est la liste d'instructions qui seront appliquées aux données (on fait les calculs, on crée des points, des droites, des cercles etc.)
- La **sortie des résultats** : ils seront en général affichés à l'écran à la fin ou petit à petit en cours de traitement (ce seront des nombres, des mots, des graphiques, des listes de nombres etc.).



EXEMPLE 1

Un algorithme demande à l'utilisateur deux nombres et affiche leur moyenne. Sa structure est la suivante :

Demander deux nombres à l'utilisateur : ce sont les entrées.
 Calculer la moyenne des deux nombres : c'est le traitement qui sera effectué par la machine.
 Afficher le résultat obtenu : c'est la sortie qui permet de communiquer le résultat à l'utilisateur.

2 Notion de variable

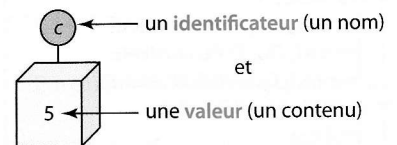
L'ordinateur doit stocker les données entrées par l'utilisateur et les résultats éventuels de ses calculs dans des zones de sa mémoire, et il doit savoir où il les a rangés !

Il repère donc chaque zone par une « adresse » en lui donnant un nom.

Variable : on peut s'imaginer une variable comme un espace de la mémoire,
 – qui est désigné par un nom, qu'on choisit. Ce nom est appelé étiquette ou **identificateur** ;
 – qui peut contenir une « valeur » (un nombre, un mot, une liste de nombres, etc.).

Affectation : une affectation est l'attribution d'une valeur (d'un contenu) à une variable.

La variable c



La valeur 5 a été **affectée** à la variable c.

Dans certains langages, il est obligatoire de déclarer les variables au début du programme en donnant leurs noms, voire en indiquant si elles vont contenir des nombres, des mots, etc.

3 Affectation et Sorties

Exemple 1 page 5 : un algorithme demande à l'utilisateur deux nombres et affiche leur moyenne.

Algorithme 1

VARIABLES : a, b, m nombres
ENTRÉES : Saisir a
 Saisir b
TRAITEMENT : m prend la valeur $(a + b)/2$
SORTIE : Afficher m

Tableau d'état des variables

Pour comprendre un algorithme, on peut exécuter les instructions pas à pas et faire un tableau d'état des variables en choisissant des valeurs pour les entrées et en indiquant à chaque ligne les contenus des variables.

Par exemple, si on entre 5 pour valeur de a et 8 pour valeur de b :

Instructions	Contenu des variables		
Saisir a	$a : 5$		
Saisir b	$a : 5$	$b : 8$	
m prend la valeur $(a + b)/2$	$a : 5$	$b : 8$	$m = 6,5$

Remarque : « m prend la valeur $(a + b)/2$ » se dit aussi : « affecter $(a + b)/2$ à m ».

Bilan

Saisir a (ou demander a , lire a) permet de demander à l'utilisateur d'introduire une donnée.

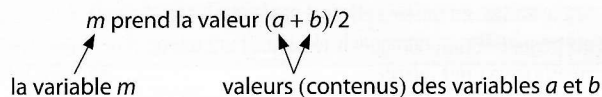
Cette instruction joue un double rôle :

- elle crée la variable nommée a ;
- elle lui affecte la valeur entrée par l'utilisateur.

Elle provoque l'arrêt de l'algorithme dans l'attente d'une valeur entrée au clavier par l'utilisateur.

m prend la valeur $(a + b)/2$ permet de déclencher le calcul de $(a + b)/2$ avec les valeurs de a et b , puis d'affecter à la variable m le résultat de ce calcul.

Important ! Noter la dissymétrie de l'affectation



Afficher m permet d'afficher à l'écran la valeur (le contenu) de la variable m .

Remarque : quand on affecte une valeur à une variable, elle remplace la valeur contenue précédemment dans la variable s'il y en avait une.

On écrit ensuite l'algorithme dans un langage de programmation qui dépend du logiciel ou de la calculatrice utilisé(e) :

Algobox	TI 82(83, 84) Plus	Python (version 3)
<pre> ▼ VARIABLES ├─ a EST_DU_TYPE NOMBRE ├─ b EST_DU_TYPE NOMBRE └─ m EST_DU_TYPE NOMBRE ▼ DEBUT_ALGORITHME ├─ LIRE a ├─ LIRE b ├─ m PREND_LA_VALEUR (a+b)/2 ├─ AFFICHER m └─ FIN_ALGORITHME </pre>	<pre> : Input "A="; A : Input "B="; B : (A+B)/2 → M : Disp M </pre>	<pre> a=float(input('a=')) b=float(input('b=')) m=(a+b)/2 print(m) </pre>
<p>Xcas (Xcasfr)</p> <pre> saisir(a); saisir(b); m:=(a+b)/2; afficher(m); </pre>	<p>Casio Graph 35+</p> <pre> "A="?→A↵ "B="?→B↵ (A+B)÷2→M↵ M↵ </pre>	<p>Scilab</p> <pre> 1 a=input("a="); 2 b=input("b="); 3 m=(a+b)/2; 4 disp(m); </pre>

Exercice résolu 1

Passer du langage naturel à l'écriture d'un algorithme avec variables et affectation

ÉNONCÉ 1. Appliquer le « programme de calcul » ci-contre plusieurs fois. Qu'observez-vous ?

2. Écrire un algorithme correspondant à ce « programme de calcul » de façon formalisée.

SOLUTION

1. Pour $a = 3$, $b = 8$ on obtient 38. Pour $a = 2$ et $b = 1$ on obtient 21. Il semble que l'on obtient le nombre formé de a dizaines et b unités.

2.

VARIABLES : a, b, c nombres
ENTRÉES : Saisir a
 Saisir b
TRAITEMENT : c prend la valeur $5 \times a$
 c prend la valeur $c + 7$
 c prend la valeur $2 \times c$
 c prend la valeur $c + b$
 c prend la valeur $c - 14$
SORTIE : Afficher c

Choisir deux entiers a et b entre 1 et 9.
Multiplier a par 5.
Ajouter 7.
Doublé le résultat obtenu.
Ajouter b .
Soustraire 14.

Exercices d'application

Corrigés sur le site.

1 On considère l'algorithme suivant écrit en langage naturel :

« Choisir un nombre entier, lui retrancher 6. Multiplier le résultat obtenu par le nombre de départ. Ajouter 9 à ce produit. »

1. Exécuter cet algorithme pour plusieurs entiers. Qu'observe-t-on ?

2. Écrire cet algorithme sous forme formalisée (avec entrées, traitement, sorties, variables et affectations).

2 Modifier l'algorithme 1 de la page 6 pour affecter le premier nombre du coefficient 2.

3 On considère les algorithmes suivants :

VARIABLE : a nombre
ENTRÉE : Saisir a
TRAITEMENT :
 c prend la valeur $a - 2$
 c prend la valeur c^2
 c prend la valeur $c + 1$
SORTIE : Afficher c

VARIABLE : a nombre
ENTRÉE : Saisir a
TRAITEMENT :
 c prend la valeur $a - 4$
 c prend la valeur ac
 c prend la valeur $c + 5$
SORTIE : Afficher c

Les faire tourner « à la main » ou après les avoir implémentés sur votre calculatrice avec plusieurs valeurs de a . Qu'observe-t-on ? Justifier.

4 On considère l'algorithme suivant :

VARIABLES : n, a, b, p nombres
ENTRÉE : Saisir n
TRAITEMENT : a prend la valeur $n - 6$
 b prend la valeur $n - 2$
 p prend la valeur ab
SORTIE : Afficher p

1. Montrer que si on choisit 8 comme nombre de départ le programme donne 12 comme résultat.

2. Vrai ou faux ?

Pour chacune des affirmations suivantes indiquer si elle est vraie ou fausse.

Proposition 1 : le programme peut donner un résultat négatif.

Proposition 2 : si on choisit $1/2$ comme nombre de départ le programme donne $33/4$ comme résultat.

Proposition 3 : le programme donne zéro comme résultat pour exactement deux nombres.

Proposition 4 : la fonction qui, au nombre de départ, associe le résultat du programme est une fonction linéaire.

D'après DNB 2014

5 Écrire un algorithme qui demande le nombre n d'objets achetés et affiche le prix TTC sachant qu'un objet coûte 3 euros hors-taxes et que la TVA est de 20 %.

4 Structure alternative : « Si... Alors... Sinon »

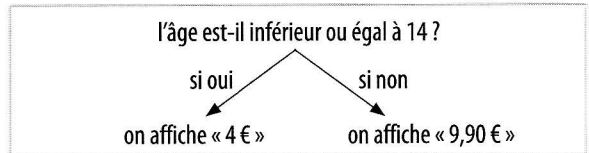
EXEMPLE 2

Le prix d'une place de cinéma pour un enfant de 14 ans ou moins est 4 € alors que pour un adulte il est 9,90 €. On veut écrire un algorithme qui demande l'âge de la personne et affiche le prix de la place. On traduit cette alternative par une instruction Si... Alors... Sinon... que l'on peut illustrer par un schéma.

Algorithme 2

```
VARIABLE : âge nombre
ENTRÉE : Saisir âge
TRAITEMENT et SORTIE :
Si âge ≤ 14 Alors Afficher « 4 € »
| Sinon Afficher « 9,90 € »
FinSi
```

Aide : on représente parfois une structure alternative par un schéma comme celui ci-dessous.



Bilan

Une structure alternative s'exprime à l'aide d'un Si... Alors... Sinon

```
Si condition Alors suite d'instructions 1
| Sinon suite d'instructions 2
FinSi
```

La condition ne doit offrir que deux réponses : vraie ou fausse !

Quand la condition est vraie, on exécute la suite d'instructions 1 ; quand elle est fausse, on exécute la suite d'instructions 2.

Quand il n'y a pas d'instruction à effectuer dans le cas où la condition est fausse, on écrit simplement :

```
Si condition Alors suite d'instructions 1
FinSi
```

Si la condition est vraie, la suite d'instructions 1 est exécutée. Si la condition est fausse, on passe à la suite de l'algorithme.

La structure alternative de l'algorithme 2 ci-dessus dans différents langages (programmes complets sur le site) [WWW](#)

Algobox	TI 82(83, 84) Plus	Python (version 3)
<pre> ▼ SI (age <= 14) ALORS DEBUT_SI AFFICHER "4 Euros" FIN_SI ▼ SINON DEBUT_SINON AFFICHER "9,90 Euros" FIN_SINON </pre>	<pre> : Input "A=", A : If A ≤ 14 : Then : Disp "4E" : Else : Disp "9.9E" : End </pre>	<pre> age=int(input('age=')) if age <= 14: print("4 Euros") else: print("9,90 Euros") </pre>
Xcas	Casio Graph 35+	Scilab
<pre> si age <= 14 alors afficher("4 Euros"); sinon afficher ("9,9 Euros") fsi :; </pre>	<pre> "A="?→A# If A≤14# Then "4E", Else "9.9E", IfEnd </pre>	<pre> 1 age=input("age="); 2 if age<=14 then 3 ---disp("4·Euros") 4 else 5 ---disp("9,90·Euros") 6 end </pre>

Exercice résolu 2

Utiliser une structure « Si... Alors... Sinon... »

ÉNONCÉ Un site de tirages photos propose des tirages au format 10 x 13 au prix de 0,15 € par photo jusqu'à 30 photos puis à 0,12 € la photo supplémentaire.

1. Quel est le prix à payer pour 20 photos ? 40 photos ?
2. Écrire un algorithme qui demande le nombre n de photos et qui affiche le prix p (en euros) à payer.

SOLUTION

1. Pour 20 photos, il faudra payer $0,15 \times 20$ € soit 3 €. Pour 40 photos, il faudra payer $0,15 \times 30$ € soit 4,50 € pour les 30 premières photos puis $0,12 \times 10$ € soit 1,20 € pour les 10 photos supplémentaires, donc en tout 5,70 €.

2.

VARIABLES : n, p nombres
 ENTRÉE : Saisir n
 TRAITEMENT : Si $n \leq 30$ Alors p prend la valeur $0,15 \times n$
 | Sinon p prend la valeur $0,15 \times 30 + 0,12 \times (n - 30)$
 FinSi
 SORTIE : Afficher « Le prix est : », p , « € »

Exercices d'application

Corrigés sur le site.

6 1. Que produit la structure alternative ci-dessous quand la pièce tombe sur Face :
 « Si la pièce tombe sur Pile Alors tu mets le couvert Sinon c'est moi » ?

2. On considère la structure alternative ci-dessous :
 « Si le dé donne 6 Alors afficher "gagné" »
 On lance un dé et on obtient 6. Que se passe-t-il ?
 Et dans le cas où on obtient 4 ?

7 On considère la structure alternative :

Si $a > b$ Alors Afficher b
 | Sinon Afficher a
 FinSi

1. Quel sera l'affichage pour :
a. $a = 3, b = 5$ **b.** $a = 2$ et $b = 0,5$ **c.** $a = 4$ et $b = 4$?
2. Qu'affiche cet algorithme pour les nombres entrés ?

8 L'IMC (Indice de Masse Corporelle) est donné par la formule suivante : $IMC = p/t^2$ où p désigne le poids en kilos et t la taille en mètres. Écrire un algorithme qui demande le poids et la taille d'un homme, affiche son IMC et ajoute un commentaire sachant que :

IMC chez l'homme	
Maigre	< 20
Poids normal	20 à 25
Surcharge pondérale	> 25

9 Lorsqu'une moquette est posée par l'entreprise elle bénéficie d'une TVA réduite : 10 % au lieu de 20 %. Recopier et compléter l'instruction suivante qui à partir du prix hors-taxes p_1 , calcule le prix TTC, p_2 au m^2 .

Si ... Alors p_2 prend la valeur $1,1 \times p_1$
 | Sinon p_2 prend la valeur ...
 FinSi

10 Écrire un algorithme demandant les coordonnées x_M et y_M d'un point M, les coefficients a et b d'une droite d d'équation $y = ax + b$ et affichant si le point M appartient ou non à la droite.

11 M. Martin utilise sa voiture personnelle pour ses déplacements professionnels. Il peut déclarer pour ses impôts les frais kilométriques F en euros en fonction du nombre d de kilomètres parcourus dans l'année, ainsi :

Si $d \leq 5\,000$ Alors F prend la valeur $d \times 0,536$
 | Sinon
 | Si $d \leq 20\,000$ Alors F prend la valeur $1\,180 + (d \times 0,30)$
 | | Sinon F prend la valeur $d \times 0,359$
 FinSi
 FinSi

1. Calculer le montant des frais F pour une distance de :
a. 3 000 km **b.** 10 000 km **c.** 25 000 km.
2. Illustrer ces structures alternatives par un schéma.

5 Boucles « Pour... » : quand on connaît le nombre de répétitions

EXEMPLE

On place une somme S au taux annuel de 4 % pendant 10 ans. Chaque année la somme placée augmente de 4 % donc est multipliée par $1 + \frac{4}{100} = 1,04$. On voudrait connaître la somme obtenue au bout de 10 ans.

Plutôt que d'écrire 10 fois de suite l'instruction « multiplier par 1,04 », on utilise une boucle « Pour ... Fin Pour ».

Algorithme 3

VARIABLE : S nombre, i nombre

ENTRÉE : Saisir S

TRAITEMENT :

```

Pour  $i$  allant de 1 à 10 Faire
|    $S$  prend la valeur  $S \times 1,04$ 
FinPour
    
```

SORTIE : Afficher S

Explications de l'algorithme pas à pas

L'utilisateur entre un nombre, par exemple 4 000. S contient la valeur 4 000.

On entre dans la boucle « Pour... FinPour »

1^{er} passage : i prend la valeur 1 ; S prend la valeur $4\,000 \times 1,04 = 4\,160$

2^e passage : i prend la valeur 2 ; S prend la valeur $4\,160 \times 1,04 = 4\,326,4$

⋮

10^e passage : i prend la valeur 10 ; S prend la valeur 5 920,98 (à 0,01 près)

La boucle est finie puisque i contient 10. On effectue l'instruction suivante.

La valeur contenue dans S est affichée.

Dans l'exemple ci-dessus on dit que l'on passe 10 fois dans la boucle ; i est le **compteur** de la boucle. Il est **incrémenté** (augmenté) de 1 à chaque passage dans la boucle. On peut lui donner un autre nom que i . Il pourrait aller de 4 à 200, ou même de 1 à n où n est une variable contenant un nombre entier (au moins égal à 1).

Bilan Écriture d'une boucle, le nombre de répétitions étant connu

La répétition d'une suite d'instructions un certain nombre de fois s'appelle une **boucle** ou une **structure itérative**.

Une répétition 100 fois d'une suite d'instructions s'écrit de la façon ci-contre :

```

Pour  $i$  allant de 1 à 100 Faire
|   Suite d'instructions
FinPour
    
```

Les programmes écrits à partir de l'algorithme 3 dans différents langages [WWW](#)

Algobox	TI 82(83, 84) Plus	Python (version 3)
<pre> VARIABLES ├── S EST_DU_TYPE NOMBRE ├── i EST_DU_TYPE NOMBRE └── DEBUT_ALGORITHME ├── LIRE S └── POUR i ALLANT_DE 1 A 10 ├── DEBUT_POUR ├── S PREND_LA_VALEUR S*1.04 ├── FIN_POUR └── AFFICHER S FIN_ALGORITHME </pre>	<pre> :Input "S=",S :For(I,1,10) :S*1.04→S :End :Disp S </pre>	<pre> S=float(input('S=')) for i in range(1,11): S=S*1.04 print("S=",S) </pre> <p>Attention !</p> <ul style="list-style-type: none"> • Range(1, 11) impose à i de prendre les valeurs 1, 2, ... 10. • Le retour en début de ligne marque la sortie de la boucle (sans FinPour).
<pre> saisir(S); pour k de 1 jusque 10 faire S:=S*1.04; fpour; afficher(S); </pre> <p>Attention. Le nom i est réservé et ne peut être utilisé.</p>	<pre> " S = ? " → S ← For 1 → I To 10 ← S × 1.04 → S ← Next ← S ← </pre>	<pre> 1 S=input("S="); 2 for i=1:10 3 S=S*1.04; 4 end 5 disp(S); </pre>

Exercice résolu 3

Simuler 100 lancers de dé

ÉNONCÉ On notera $NbAlea(1; 6)$ le tirage d'un nombre aléatoire entier entre 1 et 6. Écrire un algorithme qui simule 100 lancers d'un dé et qui compte le nombre de 6 obtenus.

SOLUTION

Analyse : il faut répéter 100 fois le lancer d'un dé. C'est une boucle « Pour ... FinPour ». Il faut aussi créer une variable (compteur) qui compte le nombre de 6 :
 – au début elle aura la valeur 0 (on l'initialise à 0) car on n'a pas de 6 ;
 – à chaque lancer, on augmentera sa valeur de 1 si on obtient un 6.

On obtient donc l'algorithme ci-contre.

```

VARIABLES : C, i, D entiers
INITIALISATION : C prend la valeur 0 // C est le compteur
TRAITEMENT : Pour i allant de 1 à 100 Faire
                D prend la valeur NbAlea(1; 6)
                Si D = 6 Alors C prend la valeur C + 1
                FinSi
            FinPour
SORTIE : Afficher C
    
```

Exercices d'application

Corrigés sur le site.

12 TIGE Implémenter l'algorithme 3 de la page 10 sur le logiciel ou la calculatrice de votre choix. Quel affichage obtient-on en entrant 6 000 pour S ?

13 QCM Choisir la bonne réponse. Dans l'algorithme ci-dessous :

```

VARIABLES : i, x nombres
TRAITEMENT et SORTIE : Pour i allant de 0 à 20 Faire
                        | x prend la valeur i2
                        | Afficher x
                    FinPour
    
```

1. On n'affiche que le carré de 20.
2. On passe 20 fois dans la boucle.
3. On affiche la somme des carrés des entiers de 0 à 20.
4. On affiche les carrés des entiers naturels de 0 à 20.

14 On a modifié l'algorithme 3 de la page 10 en l'algorithme suivant.

```

VARIABLES : S, i nombres
ENTRÉE : Saisir S
TRAITEMENT et SORTIE : Pour i allant de 1 à 10 Faire
                        | S prend la valeur S × 1,04
                        | Afficher S
                    FinPour
    
```

1. Quel changement a-t-on effectué ?
 2. Expliquer pas à pas ce que fait cet algorithme.
- 15** Modifier l'algorithme de l'exercice 13 pour qu'il affiche la somme des inverses des 20 premiers entiers naturels non nuls.

16 On considère l'algorithme ci-dessous :

```

VARIABLES : i, n, u nombres
TRAITEMENT et SORTIE : u prend la valeur 2
                        Pour i allant de 1 à n Faire
                            | u prend la valeur 2 × u + 1
                        FinPour
    
```

Reproduire et compléter le tableau suivant en faisant fonctionner cet algorithme pour $n = 3$.

i	1	2	3
u			

17 Vrai ou faux ? On considère l'algorithme ci-dessous :

```

VARIABLES : S, k nombres
TRAITEMENT : S prend la valeur 0
                Pour k allant de 1 à 5 Faire
                    | S prend la valeur S + k
                FinPour
SORTIE : Afficher S
    
```

1. L'instruction S prend la valeur $S + k$ est exécutée 4 fois.
2. À la fin de l'exécution de l'algorithme, la variable k contient 6.
3. L'algorithme affiche 15.

- 18** 1. Modifier l'algorithme de l'exercice 17 pour qu'il demande un entier n , ($n \geq 2$) et qu'il calcule la somme $1 + 2 + \dots + n$.
2. **TIGE** Implémenter l'algorithme obtenu sur le logiciel de votre choix et l'exécuter pour obtenir la somme $1 + 2 + 3 + \dots + 1\,000$ des 1 000 premiers entiers strictement positifs.

6 Boucles « Tant que... » : quand on connaît un test d'arrêt

EXEMPLE

On dépose sur un livret d'épargne une somme inférieure ou égale à 5 000 €.

Elle augmente chaque année de 4 % ce qui revient à dire qu'elle est multipliée chaque année par $1 + \frac{4}{100} = 1,04$.

On veut savoir au bout de combien d'années elle aura dépassé 6 000 €.

Il s'agit encore d'une boucle mais on ne connaît pas le nombre de passages dans la boucle. On connaît la condition d'arrêt : que la somme dépasse 6 000 €.

On va donc répéter l'augmentation tant que la somme reste inférieure ou égale à 6 000 €.

Il faut de plus compter le nombre d'années c'est-à-dire le nombre d'augmentations effectuées : on doit donc créer une variable qui compte le nombre de passages dans la boucle : cette variable s'appelle un **compteur**.

Algorithme 4 : Tantque...

VARIABLES : S, n nombres
 ENTRÉES : Saisir S
 INITIALISATION : n prend la valeur 0
 TRAITEMENT : **Tantque** $S \leq 6\,000$ **Faire**
 | S prend la valeur $S \times 1,04$
 | n prend la valeur $n + 1$
 FinTantque
 SORTIE : Afficher n

Explications de l'algorithme pas à pas

L'utilisateur entre un nombre, par exemple 4 000. S contient la valeur 4 000. n prend la valeur 0.

On entre dans la boucle « Tantque... FinTantque »

1^{er} passage : $S \leq 6\,000$ est vérifiée donc

S prend la valeur $4\,000 \times 1,04 = 4\,160$ et n la valeur $0 + 1 = 1$

2^e passage : $S \leq 6\,000$ est vérifiée donc

S prend la valeur $4\,160 \times 1,04 = 4\,326,4$ et n la valeur $1 + 1 = 2$

⋮

11^e passage : S contient 5 920,98 donc $S \leq 6\,000$ est vérifiée.

S prend donc la valeur $5\,920,98 \times 1,04 \approx 6\,157,82$ (à 0,01 près) et n la valeur 11.

On sort de la boucle puisque $S > 6\,000$ et on effectue l'instruction suivante.

La valeur contenue dans n , donc 11, est affichée...


Bilan Écriture d'une boucle avec un test d'arrêt

On commence par tester si la condition est vraie.

Si elle est vraie, on exécute la suite d'instructions et on recommence.

Si elle est fausse, on s'arrête et on sort de la boucle.

Tantque condition Faire
 | Suite d'instructions
 FinTantque

La boucle « Tant que... FinTantque » de l'algorithme 4 dans différents langages (programmes complets sur le site). 

Algobox	TI 82(83, 84) Plus	Python (version 3)
<pre> ▼ TANT_QUE (S<=6000) FAIRE -DEBUT_TANT_QUE -S PREND_LA_VALEUR S*1.04 -n PREND_LA_VALEUR n+1 -FIN_TANT_QUE </pre>	<pre> :While S<=6000 :S*1.04→S :N+1→N :End </pre>	<pre> while S<=6000: S=S*1.04 n=n+1 </pre>
<pre> Xcas tantque S<=6000 faire S:=S*1.04; n:=n+1; ftantque; </pre>	<pre> Casio Graph 35+ While S<=6000⇐ S×1.04→S⇐ N+1→N⇐ WhileEnd⇐ </pre>	<pre> Scilab 2 while S<=6000 3 S=S*1.04; 4 n=n+1; 5 end </pre>

Exercice résolu 4

Répéter des lancers tant qu'une condition n'est pas réalisée

ÉNONCÉ On notera $NbAlea(1; 6)$ le tirage d'un nombre aléatoire entier entre 1 et 6.

Écrire un algorithme qui simule le lancer d'un dé jusqu'à l'apparition d'un 6 et qui affiche le nombre de lancers effectués pour obtenir ce premier 6.

SOLUTION

Analyse : Il faut répéter le lancer d'un dé.

On ne connaît pas le nombre de répétitions mais on connaît la condition d'arrêt : obtenir 6.

C'est donc une boucle Tantque... FinTantque. On veut aussi obtenir l'affichage du nombre de lancers effectués donc il faut compter le nombre de passages dans la boucle. On va créer un compteur C pour cela : il sera incrémenté de 1 (augmenté de 1) à chaque passage dans la boucle. On obtient donc l'algorithme ci-contre.

Remarque : si jamais le premier lancer de dé amène un 6, on n'entre pas dans la boucle.

```
VARIABLES : C, D entiers
INITIALISATION : D prend la valeur NbAlea(1; 6)
//on effectue un premier lancer de dé
ENTRÉE : C prend la valeur 1 //C est le compteur
TRAITEMENT : Tantque D < 6 Faire
    | D prend la valeur NbAlea(1; 6)
    | C prend la valeur C + 1
FinTantque
SORTIE : Afficher C
```

Exercices d'application

Corrigés sur le site.

19 TICE Implémenter l'algorithme 4 sur le logiciel ou la calculatrice de votre choix. Quel affichage obtient-on en entrant 4 200 pour S ?

20 Dans chacun des cas suivants, dire quelle action est répétée et préciser si on connaît le nombre de répétitions ou la condition d'arrêt des répétitions.

- On tire un nombre entier au hasard entre 0 et 100 jusqu'à ce que l'on tire un multiple de 9.
- On lance 100 fois une pièce de monnaie.
- On court 1 000 m lundi, 1 500 m mardi, 2 000 m mercredi et ainsi de suite en ajoutant 500 m chaque jour jusqu'à dimanche.
- Une somme d'argent placée sur un compte est augmentée chaque année de 2,8 % pendant 15 ans.
- On lance une pièce de monnaie jusqu'à l'obtention de pile.

21 On considère l'algorithme suivant :

```
VARIABLE : n nombre entier
ENTRÉE : Demander n
TRAITEMENT : Tant que n ≥ 7 Faire
    | n prend la valeur n - 7
Fin Tant que
SORTIE : Afficher n
```

- Expliquer ce que fait cet algorithme quand on entre 25 pour valeur de n . Combien de fois passe-t-il dans la boucle ?
- Quel affichage obtient-on si on entre pour n :
a. 35 b. 248 c. 5
- Que fait cet algorithme ?

22 1. On veut faire afficher le plus grand entier naturel dont le cube est inférieur ou égal à 200. Lequel de ces algorithmes donne le résultat attendu ?

```
n prend la valeur 1
Tantque n³ ≤ 200 Faire
    | n prend la valeur n + 1
FinTantque
Afficher n
```

```
n prend la valeur 1
Tantque n³ ≤ 200 Faire
    | n prend la valeur n + 1
FinTantque
Afficher n - 1
```

2. Expliquer ce qu'affiche l'autre algorithme.

23 Voici un algorithme :

```
VARIABLES : n, a, c entiers positifs
ENTRÉE : Saisir a
TRAITEMENT : n prend la valeur 1 ; c prend la valeur 1
    Tant que c < a Faire
        | c prend la valeur c + n²
        | n prend la valeur n + 1
    Fin Tant que
SORTIE : Afficher n
```

1. Compléter le tableau suivant autant que nécessaire si on saisit 20 pour valeur de a .

a	20	20	...	
n	1	2	...	
c	1	2	...	
Test $c < a$	vrai			

2. QCM

- La sortie vaut 4.
- La sortie vaut 5.
- En fin d'algorithme la variable c contient 15.

D'après Sciences Po 2013

7 D'un langage de programmation à un autre

Les instructions

	Algobox	Xcas	Scilab	Python (version 3)
demander n et saisir n	La variable n a dû être déclarée dans les <u>variables auparavant</u> + Ajouter LIRE variable	saisir(n);	$n=\text{input}("n=")$;	si n est entier : $n=\text{int}(\text{input}('n='))$ sinon : $n=\text{float}(\text{input}('n='))$
afficher n	+ Ajouter AFFICHER Variable	afficher(n);	disp(n);	print(n)
Si ...alors ... sinon ... FinSi	+ Ajouter SI...ALORS Cocher la case « Ajouter SINON » dans la fenêtre de dialogue suivante	si ...alors ... ; sinon ... ; fsi	if ...then ... else ... end	if ... : ... else :
Pour k allant de 1 à 9 Faire ... FinPour	+ Ajouter POUR...DE...A	pour k de 1 jusque 9 faire ... ; fpour	for $k=1:9$... ; end	for k in range(1,10):
Tant que $k > 1$... FinTantQue	+ Ajouter TANT QUE...	tantque $k>1$ faire ... ; ftantque	while $k>1$... ; end	while... :

Le test d'égalité et quelques calculs

	Algobox	Xcas	Scilab	Python (version 3)
tester une égalité	==	==	==	==
tester si $a \neq b$!=	!=	<>	not $a == b$
et ; ou	$a==b$ et $a>2$ $a==b$ ou $a>2$	$(a==b)$ et $(a>2)$ $(a==b)$ ou $(a>2)$	& (pour et) (pour ou)	and (pour et) or (pour ou)
n^3	pow($n,3$)	n^3	n^3	$n^{**}3$
\sqrt{x}	sqrt(x)	sqrt(x)	sqrt(x)	from math import * sqrt(x)
reste de la division de n par 5	$n\%5$	irem($n,5$)	reste($n,5$)	$n\%5$
nombre aléatoire entre 0 et 1	random()	rand(0,1)	rand(0,1)	from random import* random()
la « partie entière » d'un nombre	floor (2.1) = 2 floor(-3.2) = -4	iPart(2.1)=2 iPart(-3.2)=-4	floor (2.1) = 2 floor(-3.2) = -4	from math import * floor(2.1) = 2 floor (-3.2) = -4