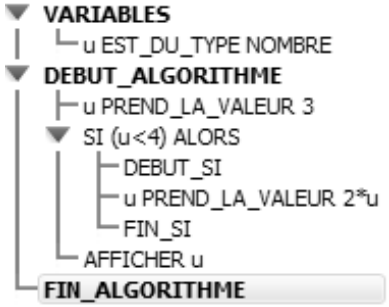
1
J'algorithme, tu algorithmes, nous algorithmons

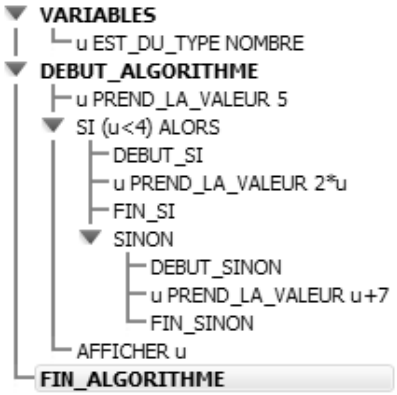
1. LES BASES

« Si… alors »

Avec le logiciel Algobox	En langage de tous les jours (!)	
	Variable	Saisir u
	Initialisation	Affecter à u la valeur 3
	Traitement	Si u < 4 alors, affecter à u la valeur 2u.
	Sortie	Afficher u

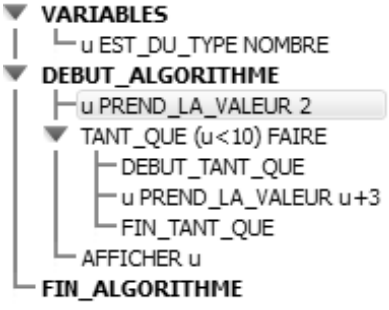
Qu'est-ce qui sort ? C'est 6 !
Cet algorithme, comme tous les autres, peut se réaliser sur les calculatrices Casio ou Texas ou encore avec le logiciel Xcas ou d'autres. Il faut alors s'adapter aux mots de chacun ou chacune.

« Si… alors… sinon »

Avec le logiciel Algobox	En langage de tous les jours	
	Variable	Saisir u
	Initialisation	Affecter à u la valeur 5
	Traitement	Si u < 4 alors, affecter à u la valeur 2u. Sinon, affecter à u la valeur u + 7
	Sortie	Afficher u

C'est le 12 qui sort.

« Tant que… »

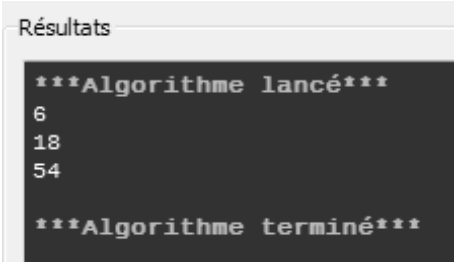
Avec le logiciel Algobox	En langage de tous les jours (!)	
	Variable	Saisir u
	Initialisation	Affecter à u la valeur 2
	Traitement	Tant que u < 10, affecter à u la valeur u + 3.
	Sortie	Afficher u

C'est le 11 qui sort.
En effet, le u de départ, u = 2, est plus petit que 10. Ce u va être changé en u + 3 = 5.
Mais ce u est encore plus petit que 10, ça dure tant que ! Alors ce u va être changé en u + 3 = 8. Encore plus petit que 10! Ce u va être changé en u + 3 = 11. C'est fini, il n'y a plus tant que puisque ce u est plus grand que 10.

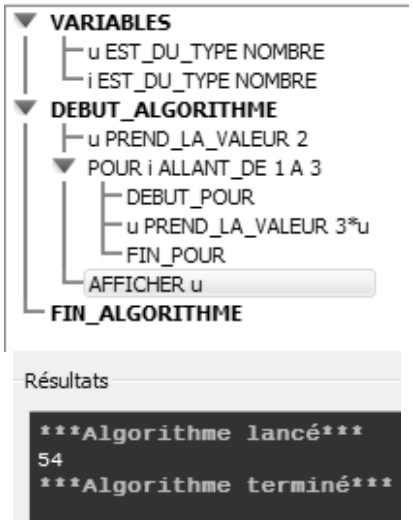
« Pour…»

Avec le logiciel Algobox	En langage de tous les jours (!)	
VARIABLES u EST_DU_TYPE NOMBRE i EST_DU_TYPE NOMBRE DEBUT_ALGORITHME u PREND_LA_VALEUR 2 POUR i ALLANT_DE 1 A 3 DEBUT_POUR u PREND_LA_VALEUR 3*u AFFICHER u FIN_POUR FIN_ALGORITHME	Variables	u, i
	Initialisation	Affecter à u la valeur 2
	Traitement	Répéter 3 fois Affecter à u la valeur 3u
	Sortie	Afficher u

La même opération : « multiplier par 3 » s'effectue en boucle 3 fois.
De u = 2 on passe à u = 6, puis de u = 6 on passe à u = 18, puis de u = 18 à u = 54.
La variable i donne le nombre de fois que l'opération doit se faire : une fois, deux fois, trois fois.
C'est très utile pour calculer les premiers termes d'une suite arithmétique ou géométrique.
Algobox affiche en sortie :

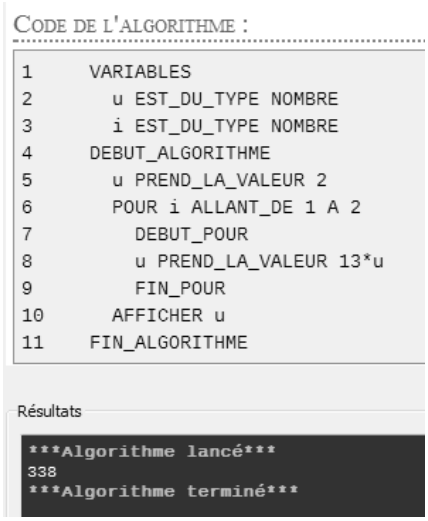


Attention ! Si on demande à afficher u uniquement après les calculs en boucle, seul le dernier u s'affichera :



« Tant que… » et « Pour… » sont des boucles

Ecouter une chanson en boucle c'est l'écouter plusieurs fois, un algorithme avec une boucle c'est une même opération qui se répète ; une itération, dit-on aussi.
On peut refaire faire la même opération par « Tant que » ou par « Pour ».
Si on veut répéter une opération un nombre de fois fixé à l'avance alors on utilisera « Pour » ; si on veut répéter une opération jusqu'à ce qu'il se passe quelque chose qu'on veut observer, alors on utilise « Tant que ».
Par exemple soit une suite géométrique de premier terme 2 et de raison 13.
On veut calculer le troisième terme, la multiplication par 13 étant répétée deux fois, alors on utilise « Pour… » :



Mais si on veut calculer le premier terme de cette suite qui dépassera 10 000, la multiplication par 13 étant répétée un nombre inconnu de fois, alors on utilise « Tant que… » :

CODE DE L'ALGORITHME :

```
1     VARIABLES
2       u EST_DU_TYPE NOMBRE
3     DEBUT_ALGORITHME
4       u PREND_LA_VALEUR 2
5       TANT_QUE (u<10000) FAIRE
6         DEBUT_TANT_QUE
7         u PREND_LA_VALEUR 13*u
8         FIN_TANT_QUE
9       AFFICHER u
10    FIN_ALGORITHME
```

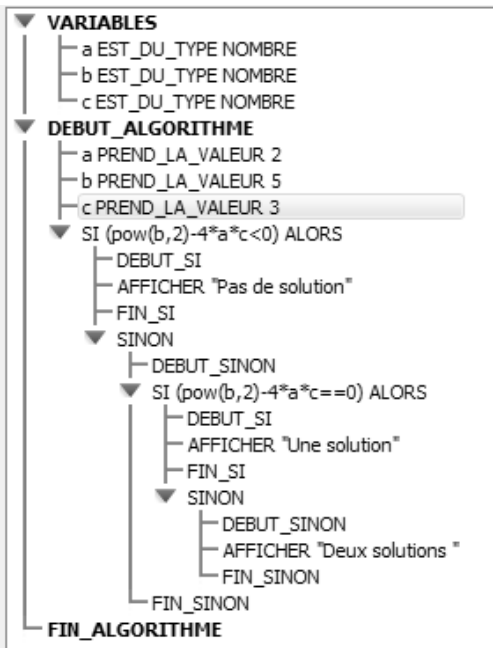
RÉSULTATS :

Résultats

```
***Algorithme lancé***
57122
***Algorithme terminé***
```

✍ Exercice d'application 1

Ci-suit un algorithme :



1. Quel sera la réponse de cet algorithme ? De quelles solutions parle-t-il ? Calculer ces solutions.

2. Déterminer le signe du trinôme $2x^2+5x+3$ selon les valeurs de x.

Corrigé

1.

- L'algorithme calcule : pow(b,2)–4*a*c = $b^2-4\times a\times c = 5^2-4\times 2\times 3 = 1$.

Le résultat est strictement plus grand que 0 ; l'algorithme dira : « Deux solutions ».

- Oui, $b^2-4\times a\times c$ est le discriminant du trinôme ax^2+bx+c.

Pour $a=2,\ b=5,\ c=3$ le discriminant Δ est strictement positif, donc l'équation $ax^2+bx+c=0 \Leftrightarrow 2x^2+5x+3=0$ a deux solutions données par les formules bien connues (qu'il faut bien connaître) :

$$\begin{cases} \text{la première solution} = \dfrac{-b-\sqrt{\Delta}}{2a} = \dfrac{-5-\sqrt{1}}{2\times 2} = -1{,}5 \\ \text{la deuxième solution} = \dfrac{-b+\sqrt{\Delta}}{2a} = \dfrac{-5+\sqrt{1}}{2\times 2} = -1 \end{cases}.$$

2. Le signe d'un trinôme à discriminant positif est donné dans le cours de première. Si la variable x est entre les deux solutions (les zéros ou racines du trinôme) alors le trinôme est du signe de –a, sinon, si la variable x est à l'extérieur des solutions, le trinôme est du signe de a.

x	$-\infty$		$-1{,}5$		-1		$+\infty$
le signe de $2x^2+5x+3$		+	0	–	0	+	

« Confirmation » → Si $x=1$,

2. DEUX ALGORITHMES DU HASARD

The first

```
A et C sont des variables.
C prend la valeur 0.
Répéter 2 fois
   A prend une valeur aléatoire entière entre 1 et 5.
   Si A > 3 alors C prend la valeur C + 1
Fin répéter
Afficher C.
```

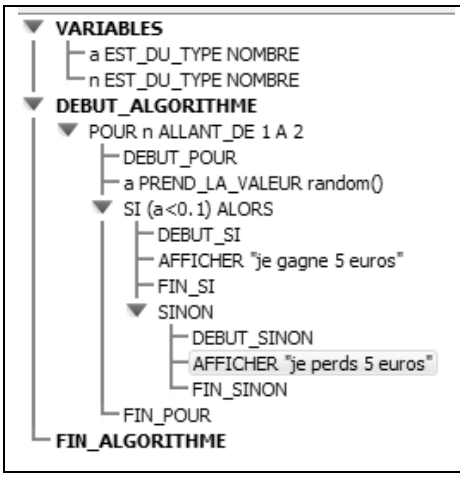
Que peut-il se passer ?
La première fois, A prend par hasard la valeur 4. Alors, comme A = 4 > 3, C qui valait 0 vaut maintenant 0 + 1 = 1.
La deuxième fois, A prend par hasard la valeur 2. Alors, comme A = 2 n'est pas plus grand que 3, C ne change pas et reste égal à 1.
L'algorithme affichera en réponse : C = 1.
Il peut se passer autre chose :

– la première fois, A prend par hasard la valeur 2. Alors, comme A = 2 ≤ 3, C qui valait 0 ne change pas et vaut toujours 0 ;

– la deuxième fois, A prend par hasard la valeur 1. Alors, comme A = 1 ≤ 3, C ne change encore pas et vaut toujours 0.

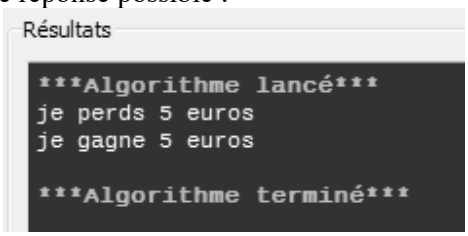
L'algorithme affichera en réponse : C = 0.
Tout tient du hasard !

The second



random() donne un nombre aléatoire entre 0 et 1.

On répète deux fois la même expérience : si la valeur de a, prise au hasard, est plus petite que 0,1 alors est affiché : « Je gagne 5 euros ». Sinon on verra affiché : « Je perds 5 euros ».
Par exemple, voilà une réponse possible :



Cette même réponse arrive grosso-modo 9 fois sur 100 ; c'est une question de loi binomiale.

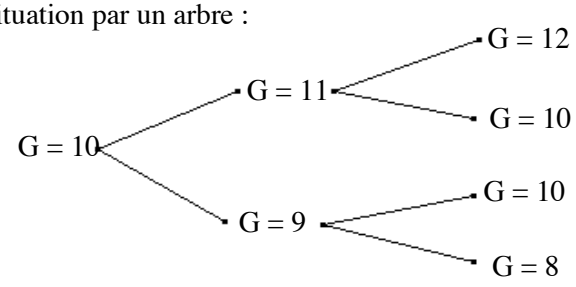
✍ Exercice d'application 2

A et G sont des variables.
G prend la valeur 10.
Répéter 2 fois
    A prend une valeur aléatoire entre 0 et 1.
    Si A < 0,4, alors G prend la valeur G – 1.
    Sinon, G prend la valeur G + 1.
Fin répéter
Afficher G.

1. Justifier que G = 8 est un résultat possible.
2. Donner la liste de tous les résultats possibles pour G.
3. Quelle est la probabilité que G soit égal à 12 à la fin ?

Corrigé

1. La première fois, imaginons que A prenne la valeur aléatoire 0,3 < 0,4 ; alors le G initial, G = 10, prend la valeur G – 1 = 9.
Enfin, la deuxième fois, imaginons que A prenne la valeur aléatoire 0,2 < 0,4 ; alors G = 9 se transforme en G – 1 = 8.
L'algorithme affichera 8 pour G.
2. Décrivons la situation par un arbre :



Les résultats possibles sont 0, 10, 12.

3. Les chances que G baisse de 1 sont de 0,4 ; c'est la probabilité que G baisse de 1 au cours d'une étape. Les chances que G augmente de 1 sont de $0{,}6 = 1 - 0{,}4$.
Pondérons l'arbre précédent.