Logique et algorithmes

I.R.E.M. La Réunion, 12 avril 2017

Brevet des collèges Polynésie, 21 juin 2016

Voici un programme de calcul :

- **Choisir un nombre entier positif**
- **Ajouter 1**
- **Calculer le carré du résultat obtenu**
- **Enlever le carré du nombre de départ.**

Brevet des collèges Polynésie, 21 juin 2016

Voici un programme de calcul :

- **Choisir un nombre entier positif**
- **Ajouter 1**
- **Calculer le carré du résultat obtenu**
- **Enlever le carré du nombre de départ.**

On applique ce programme de calcul au nombre 3.
Montrer qu'on obtient 7.

Brevet des collèges Polynésie, 21 juin 2016

Voici un programme de calcul :

- **Choisir un nombre entier positif**
- **Ajouter 1**
- **Calculer le carré du résultat obtenu**
- **Enlever le carré du nombre de départ.**

Chaque résultat peut s'obtenir en ajoutant le nombre entier de départ et le nombre entier qui le suit

Brevet des collèges Polynésie, 21 juin 2016

Voici un programme de calcul :

- Choisir un nombre entier positif
- Ajouter 1
- Calculer le carré du résultat obtenu
- Enlever le carré du nombre de départ.

à qui ?

de qui ?

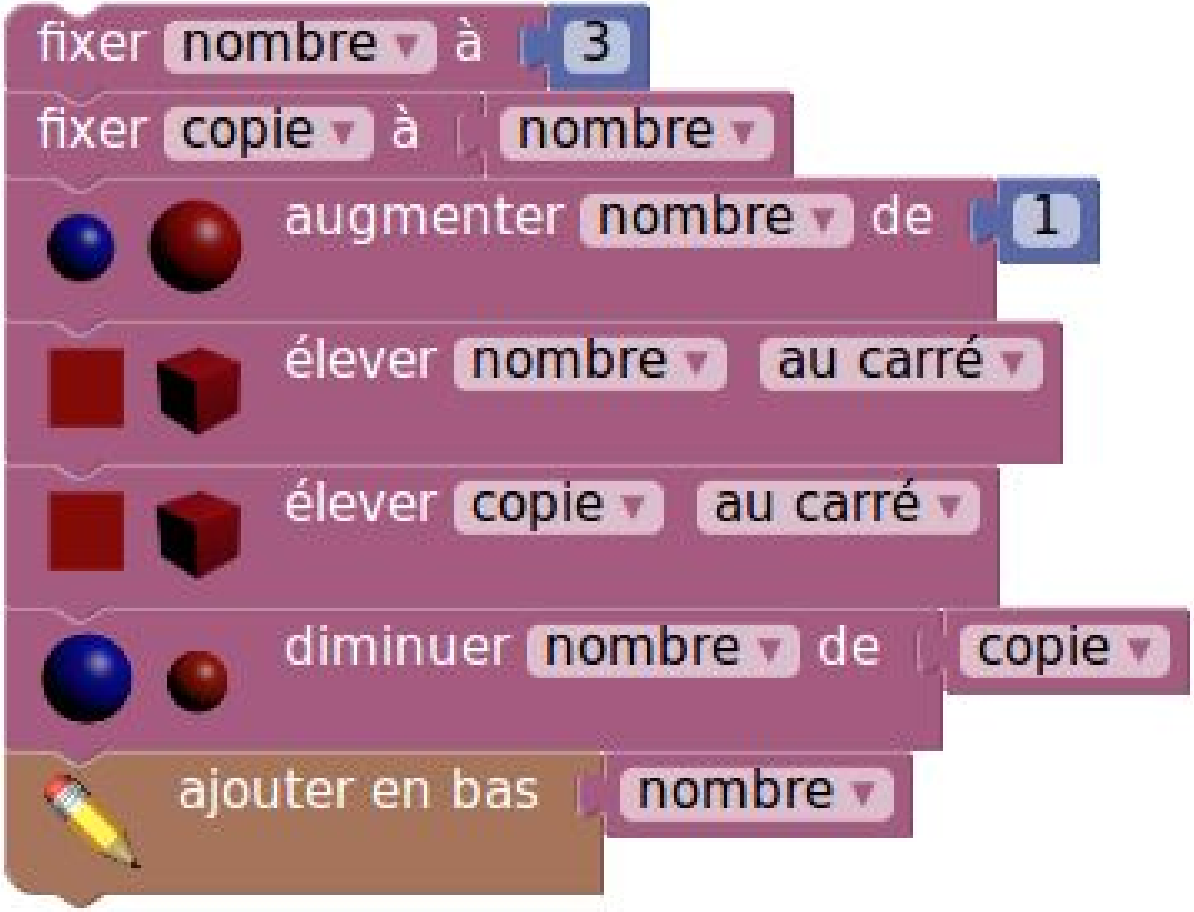
quand ?

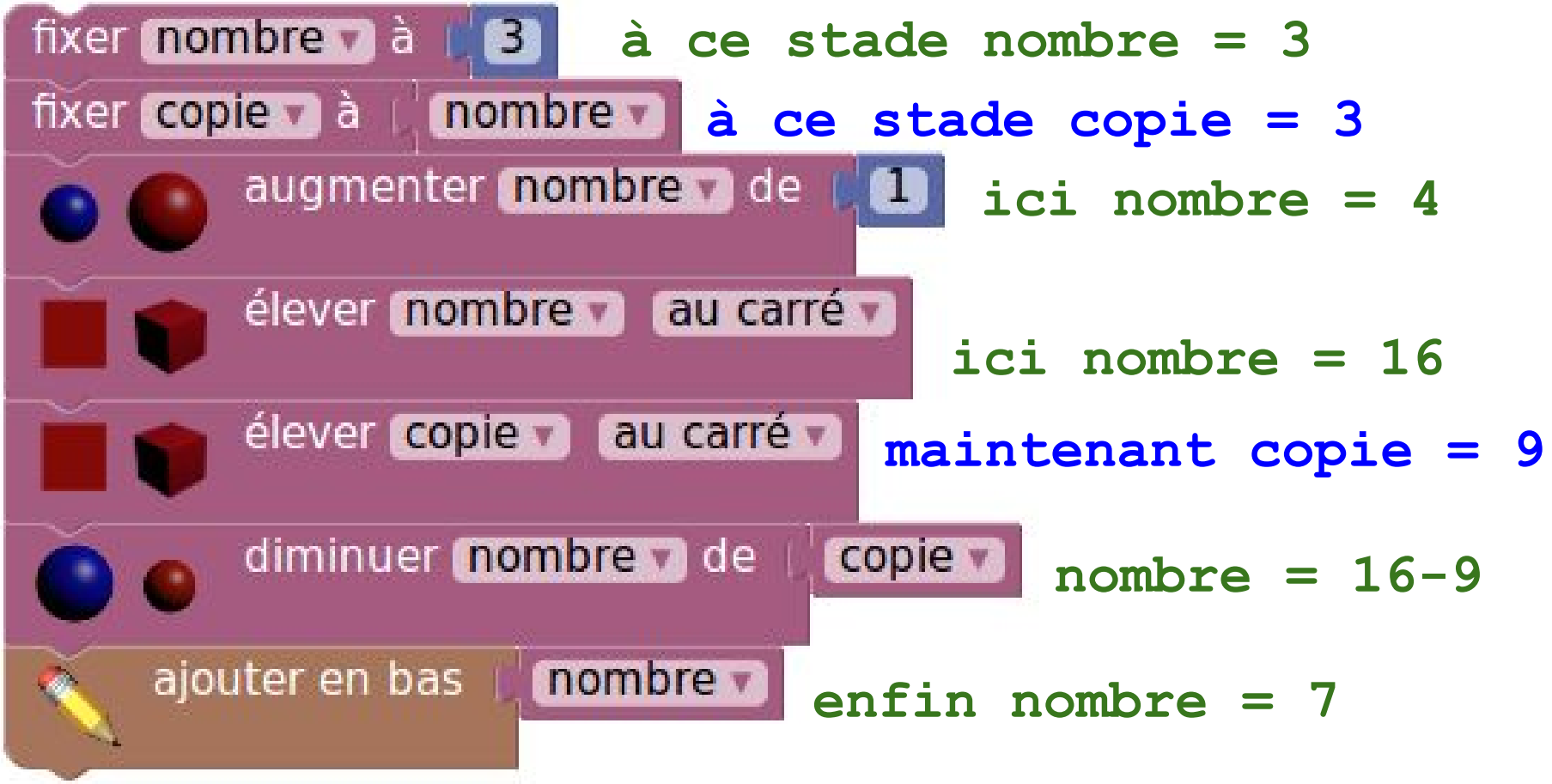
Brevet des collèges Polynésie, 21 juin 2016

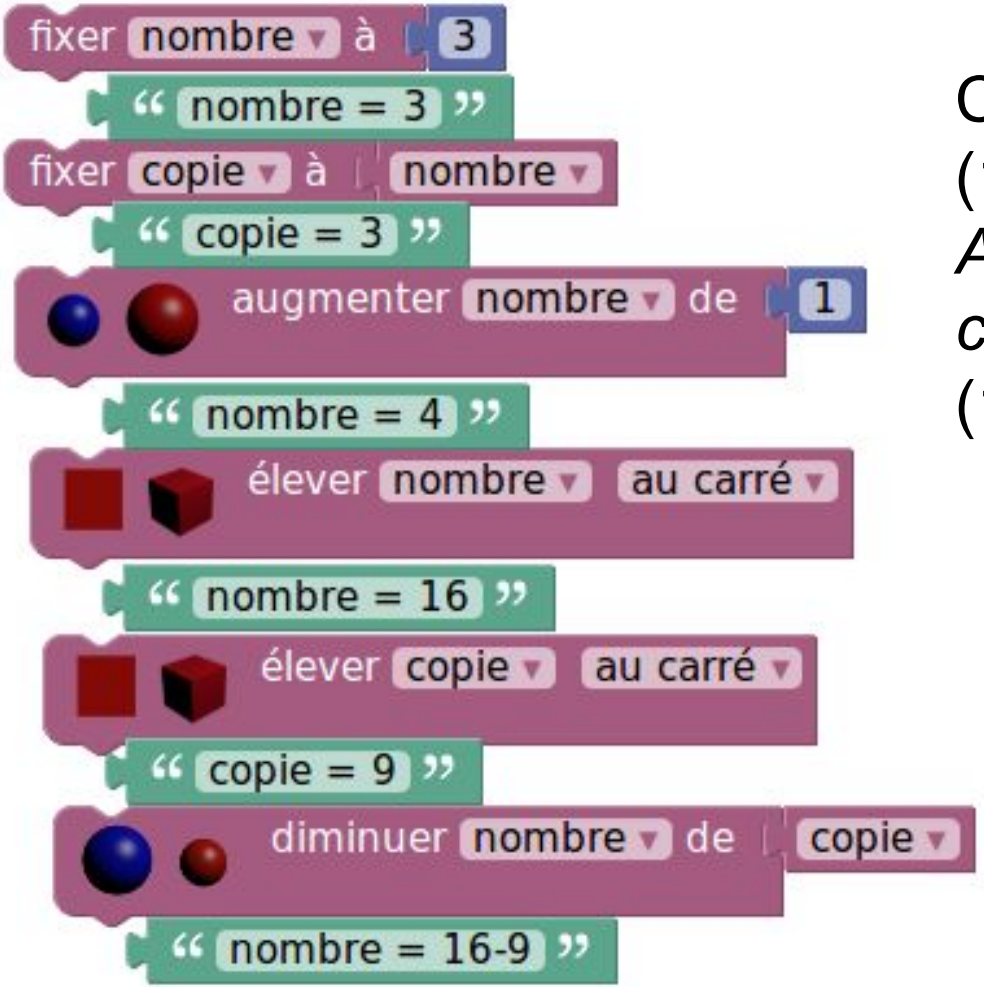
Voici un programme de calcul :

- **Choisir un nombre entier positif**
- **Ajouter 1**
- **Calculer le carré du résultat obtenu**
- **Enlever le carré du nombre de départ.**

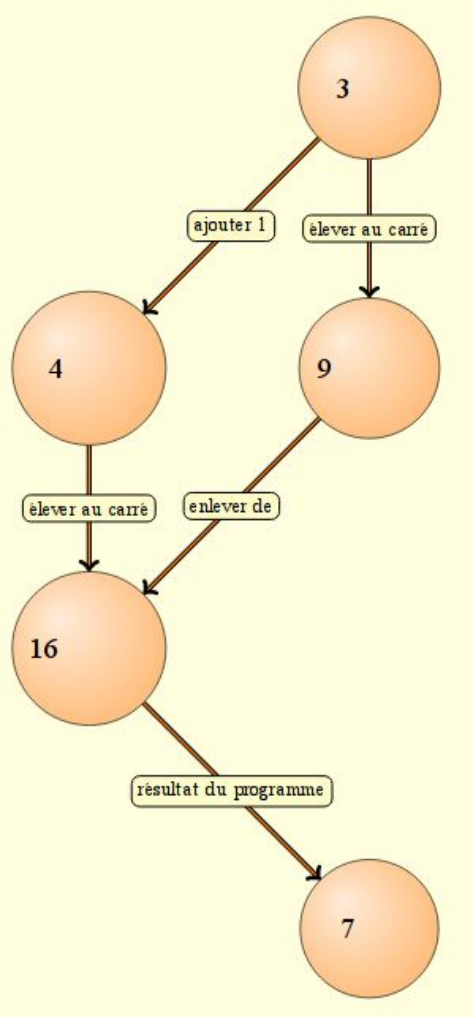
On applique ce programme de calcul au nombre 3.
Montrer qu'on obtient 7.


fixer nombre à 3
fixer copie à nombre
augmenter nombre de 1
élever nombre au carré
élever copie au carré
diminuer nombre de copie
ajouter en bas nombre


fixer nombre à 3
à ce stade nombre = 3
fixer copie à nombre
à ce stade copie = 3
augmenter nombre de 1
ici nombre = 4
élever nombre au carré
ici nombre = 16
élever copie au carré
maintenant copie = 9
diminuer nombre de copie
nombre = 16-9
ajouter en bas nombre
enfin nombre = 7



C.A.R. Hoare
(1934 -)
An axiomatic basis for computer programming
(1967)



A.M. Turing: *Checking a large routine*, 1949
J. McCarthy: *Towards a mathematical theory of computation*, 1962

Robert W. Floyd
(1936 - 2001)
assigning meanings to programs
(1967)

Ariane V le 4 juin 1996

Un dépassement de capacité a fait croire à une déviation de la trajectoire; en corrigeant cette trajectoire, l'ordinateur de bord a provoqué une accélération latérale trop importante: Boum !

RER ligne A

Une rame toutes les 2 minutes; le tout est piloté par un logiciel prouvé par Coq à l’aide de théories topologiques

Brevet des collèges Polynésie, 21 juin 2016

Voici un programme de calcul :

- **Choisir un nombre entier positif**
- **Ajouter 1**
- **Calculer le carré du résultat obtenu**
- **Enlever le carré du nombre de départ.**

Chaque résultat peut s'obtenir en ajoutant le nombre entier de départ et le nombre entier qui le suit

CoqIde

```
Definition f1 (x:nat): nat := let y:=x+1 in y*y-x*x.
Definition f2 (x: nat): nat := x+x+1.
Compute f1 3.
Compute f1 8.
Compute f1 13.
```

```
Warning: query commands should not be
inserted in scripts
     = 7
     : nat
```

CoqIde

```
Definition f1 (x:nat): nat := let y:=x+1 in y*
Definition f2 (x: nat): nat := x+x+1.
Theorem Tahiti: forall n: nat, f1(n) = f2(n).
Proof.
    unfold f1, f2.
    induction n.
    auto.
```

```
1 subgoals
____________________________________
____(1/1)
forall n : nat, (n + 1) * (n + 1)
- n * n = n + n + 1
```

CoqIde

```
Definition f1 (x:nat): nat := let y:=x+1 in y*y-x*x.
Definition f2 (x: nat): nat := x+x+1.
Theorem Tahiti: forall n: nat, f1(n) = f2(n).
Proof.
    unfold f1, f2.
    induction n.
    auto.
```

```
2 subgoal
_____________________________________(1/2)
(0 + 1) * (0 + 1) - 0 * 0 = 0 + 0 + 1
_____________________________________(2/2)
(S n + 1) * (S n + 1) - S n * S n = S n + S n + 1
```

CoqIde

```
Definition f1 (x:nat): nat := let y:=x+1 in y*y-x*x.
Definition f2 (x: nat): nat := x+x+1.
Theorem Tahiti: forall n: nat, f1(n) = f2(n).
Proof.
    unfold f1, f2.
    induction n.
    auto.
```

```
1 subgoals
n : nat
IHn : (n + 1) * (n + 1) - n * n = n + n + 1
                                          (1/1)
(S n + 1) * (S n + 1) - S n * S n = S n + S n + 1
```

Le seigneur des anneaux

```
Require Import ZArith.
Open Scope Z_scope.
Definition f1 (x:Z): Z := let y:=x+1 in y*y-x*x.
Definition f2 (x: Z): Z := x+x+1.
Theorem tahiti: forall (x: Z), f1(x) = f2(x).
Proof.
    intros.
    unfold f1, f2.
    ring.
Qed.
```

```
1 subgoals
x : Z
_____________________________________(1/1)
f1 x = f2 x
```

Le seigneur des anneaux

```
Require Import ZArith.
Open Scope Z_scope.
Definition f1 (x:Z): Z := let y:=x+1 in y*y-x*x.
Definition f2 (x: Z): Z := x+x+1.
Theorem tahiti: forall (x: Z), f1(x) = f2(x).
Proof.
    intros.
    unfold f1, f2.
    ring.
Qed.
```

```
1 subgoals
x : Z
______________________________________(1/1)
(x + 1) * (x + 1) - x * x = x + x + 1
```

Le seigneur des anneaux

```
Require Import ZArith.
Open Scope Z_scope.
Definition f1 (x:Z): Z := let y:=x+1 in y*y-x*x.
Definition f2 (x: Z): Z := x+x+1.
Theorem tahiti: forall (x: Z), f1(x) = f2(x).
Proof.
    intros.
    unfold f1, f2.
    ring.
Qed.
```

```
No more subgoals.
```

Le seigneur des anneaux

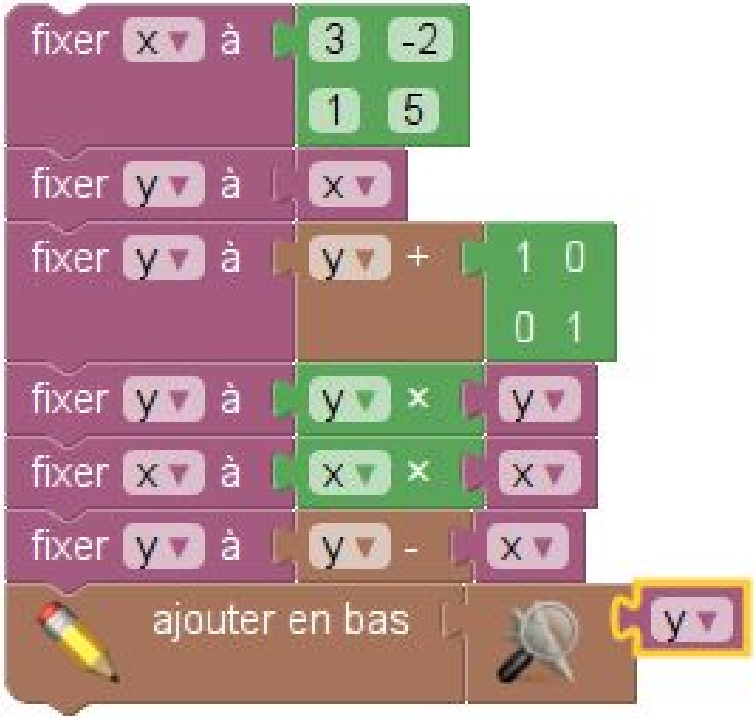
```
Require Import Reals.
Open Scope R_scope.
Definition f1 (x:R): R := let y:=x+1 in y*y-x*x.
Definition f2 (x: R): R := x+x+1.
Theorem tahiti: forall (x: R), f1(x) = f2(x).
Proof.
    intros.
    unfold f1, f2.
    ring.
Qed.
```

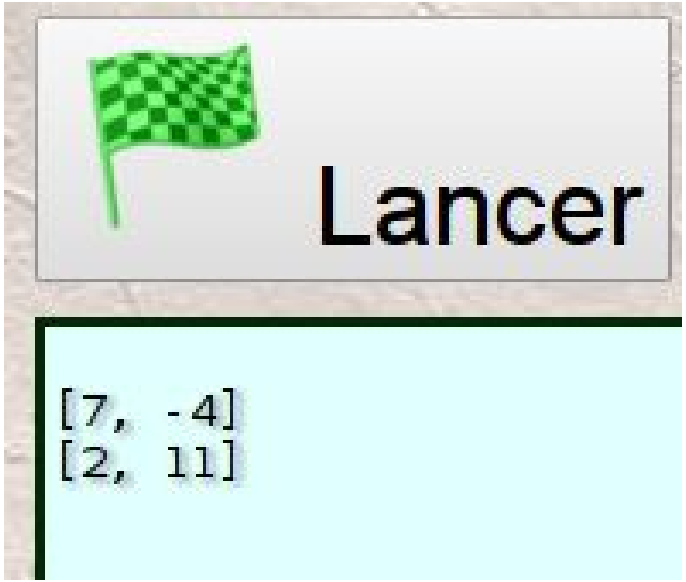
des atolls aux anneaux

Voici un programme de calcul :

- **Choisir une matrice**
- **Ajouter 1 à chacun de ses termes diagonaux**
- **Calculer le carré de la matrice obtenue**
- **Enlever le carré de la matrice de départ.**

des atolls aux anneaux





Jean-Christophe Filliâtre
(1971 -)
codéveloppeur de

- Coq
- Why

why3

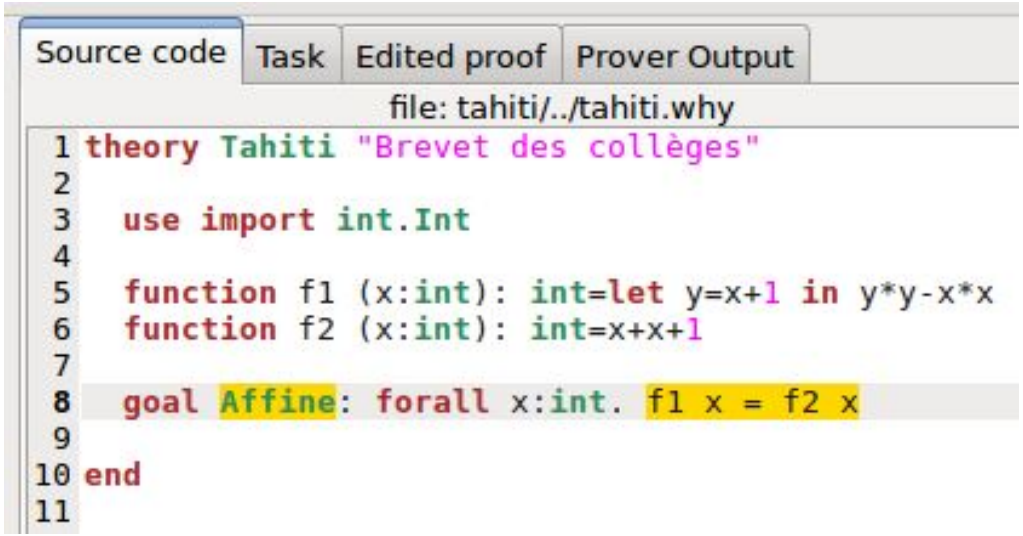
Source code | Task | Edited proof | Prover Output

file: tahiti/../tahiti.why

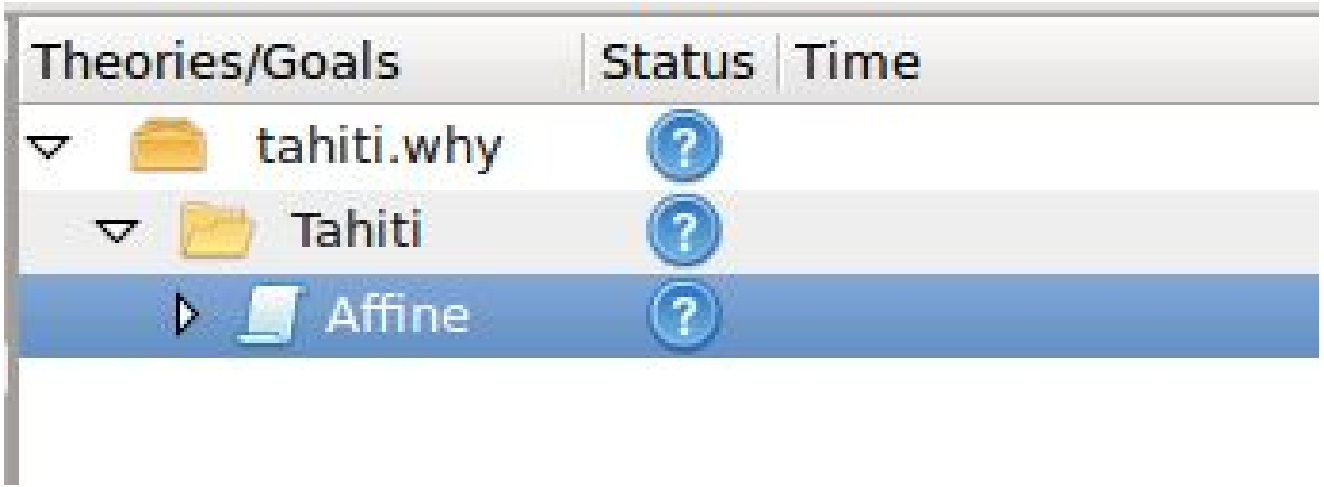
```
 1 theory Tahiti "Brevet des collèges"
 2
 3   use import int.Int
 4
 5   function f1 (x:int): int=let y=x+1 in y*y-x*x
 6   function f2 (x:int): int=x+x+1
 7
 8   goal Affine: forall x:int. f1 x = f2 x
 9
10 end
11
```

why3

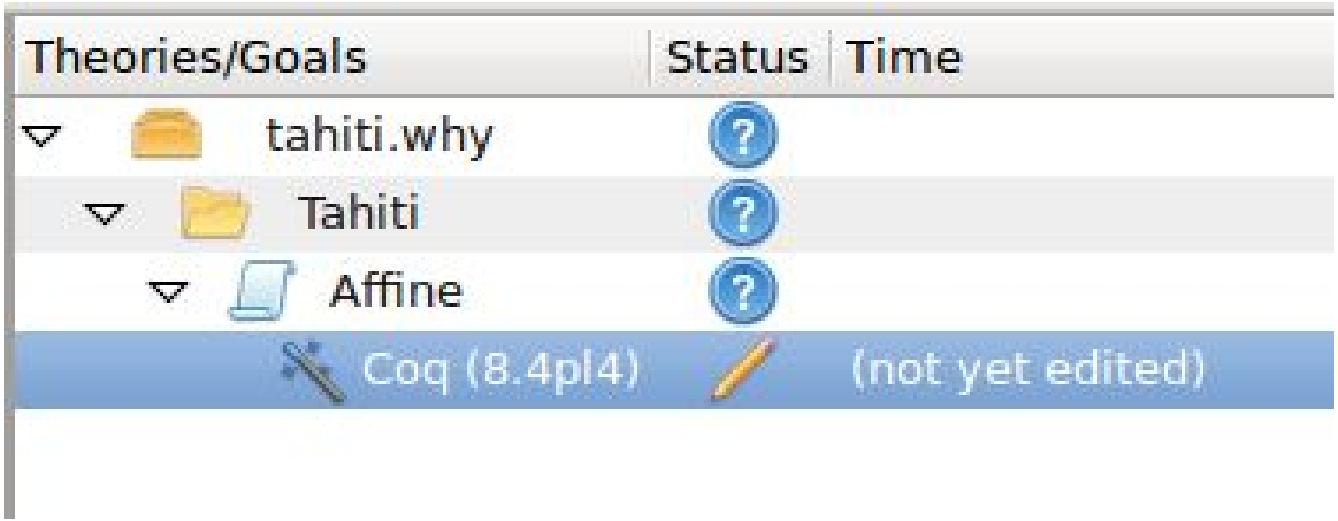

Context
Unproved goals
All goals
Strategies
Compute
Inline
Split
Provers
Alt-Ergo (0.99.1)
CVC3 (2.4.1)
Coq (8.4pl4)
Spass (3.7)


Source code
Task
Edited proof
Prover Output
file: tahiti/../tahiti.why
1 theory Tahiti "Brevet des collèges"
2
3 use import int.Int
4
5 function f1 (x:int): int=let y=x+1 in y*y-x*x
6 function f2 (x:int): int=x+x+1
7
8 goal Affine: forall x:int. f1 x = f2 x
9
10 end
11

why3



why3


Theories/Goals
Status
Time
tahiti.why
Tahiti
Affine
Coq (8.4pl4)
(not yet edited)

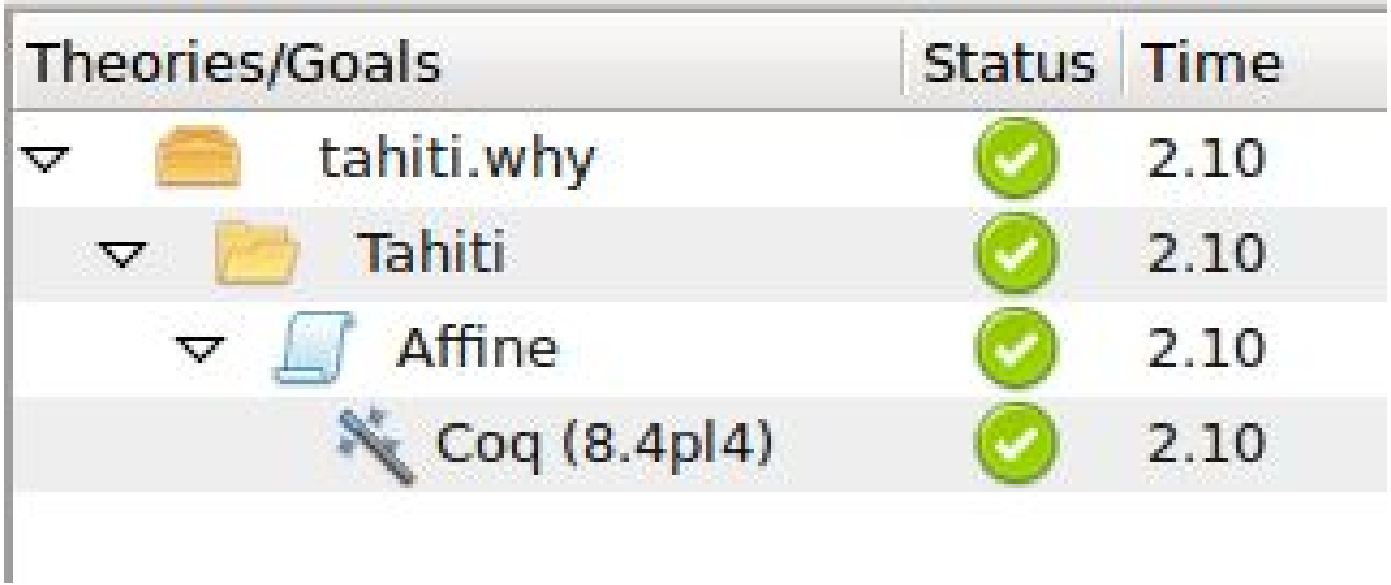
why3

```
(* This file is generated by Why3's Coq driver *)
(* Beware! Only edit allowed sections below    *)
Require Import BuiltIn.
Require BuiltIn.
Require int.Int.

(* Why3 assumption *)
Definition f1 (x:Z): Z := let y := (x + 1%Z)%Z in ((y * y)%Z - (x * x)%Z)%Z.

(* Why3 assumption *)
Definition f2 (x:Z): Z := ((x + x)%Z + 1%Z)%Z.

(* Why3 goal *)
Theorem Affine : forall (x:Z), ((f1 x) = (f2 x)).
intros x.

Qed.
```

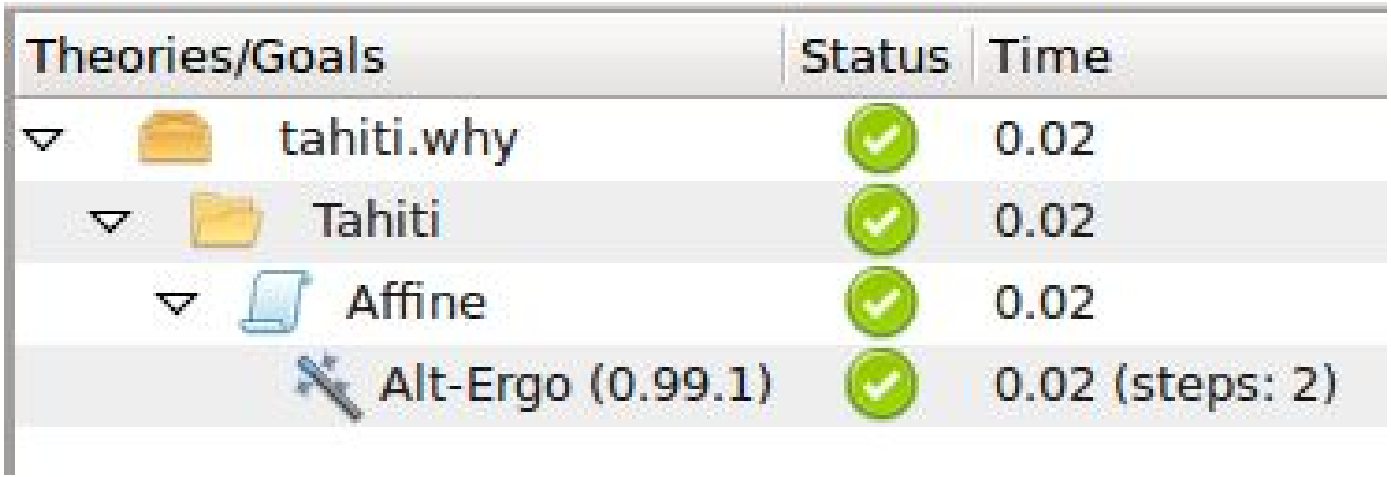
why3

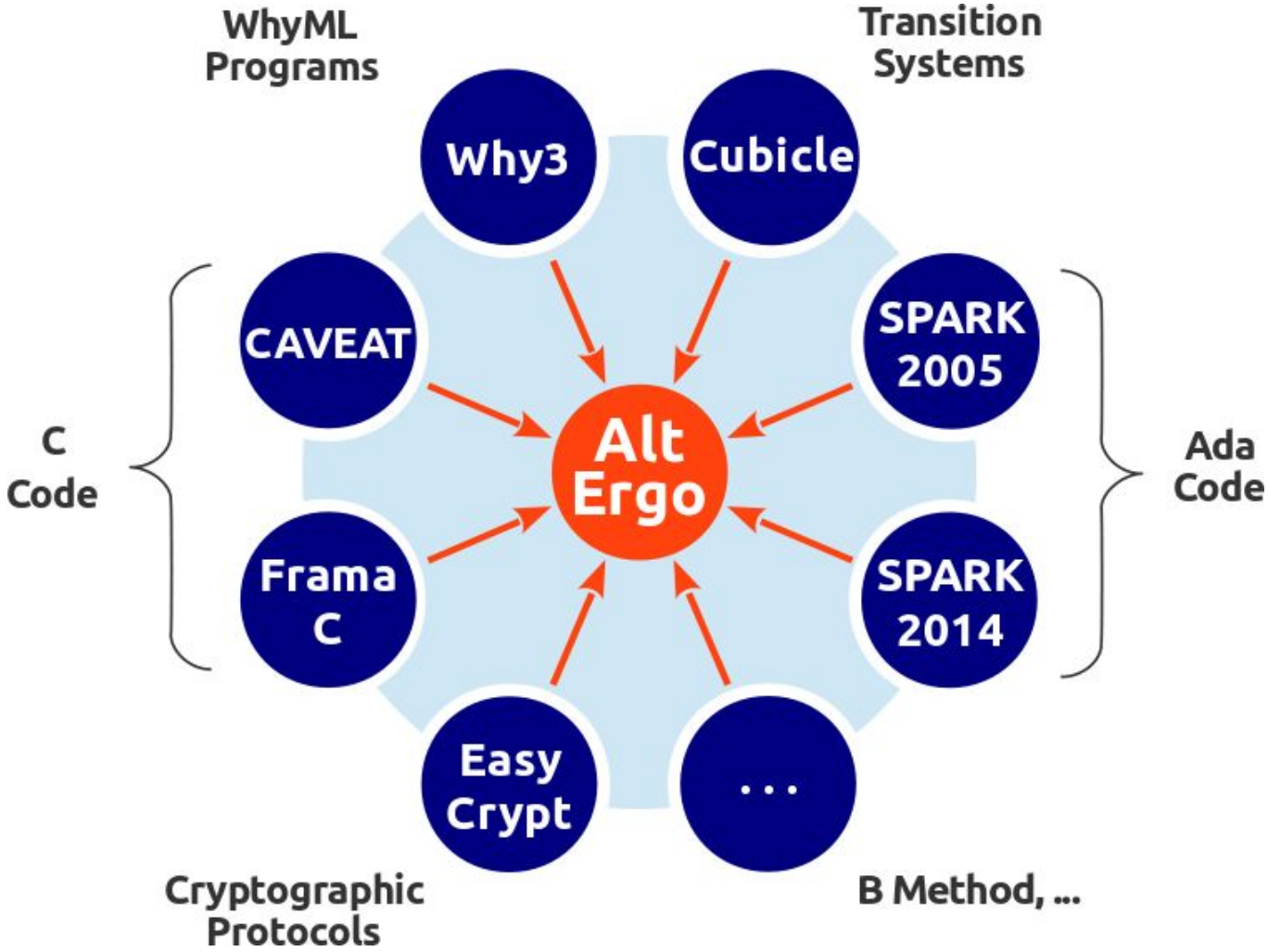
```
(* Why3 assumption *)
Definition f2 (x:Z): Z := ((x + x)%Z + 1%Z)%Z.

(* Why3 goal *)
Theorem Affine : forall (x:Z), ((f1 x) = (f2 x)).
intros x.
unfold f1, f2.
ring.

Qed.
```

why3


Theories/Goals
Status
Time
tahiti.why
2.10
Tahiti
2.10
Affine
2.10
Coq (8.4pl4)
2.10

why3

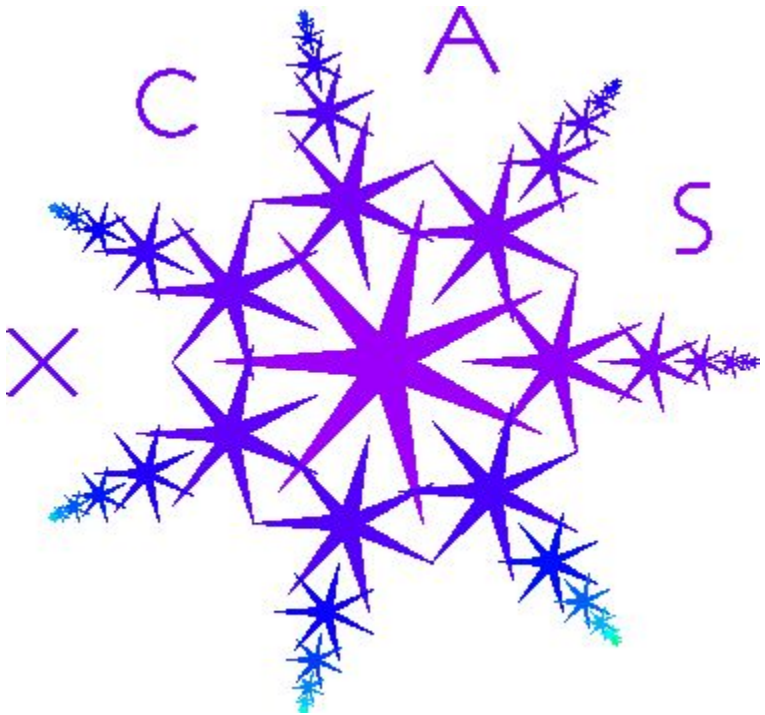

Theories/Goals
Status
Time
tahiti.why
0.02
Tahiti
0.02
Affine
0.02
Alt-Ergo (0.99.1)
0.02 (steps: 2)


WhyML
Programs
Transition
Systems
Why3
Cubicle
CAVEAT
SPARK
2005
C
Code
Alt
Ergo
Ada
Code
Frama
C
SPARK
2014
Easy
Crypt
...
Cryptographic
Protocols
B Method, ...

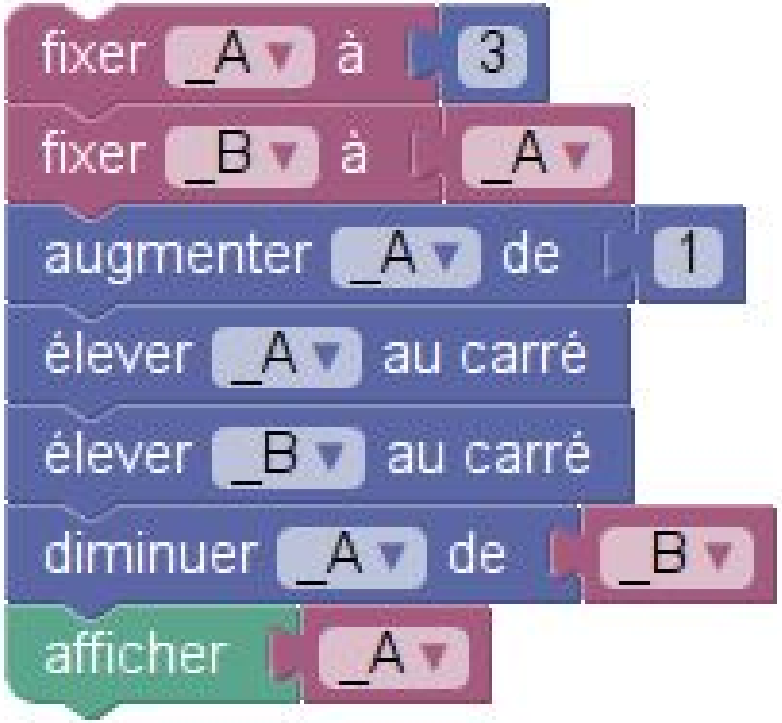
Alt-Ergo

```
goal tahiti :

  forall x,y:int.

  y=x+1   ->

  y*y-x*x=x+x+1
```

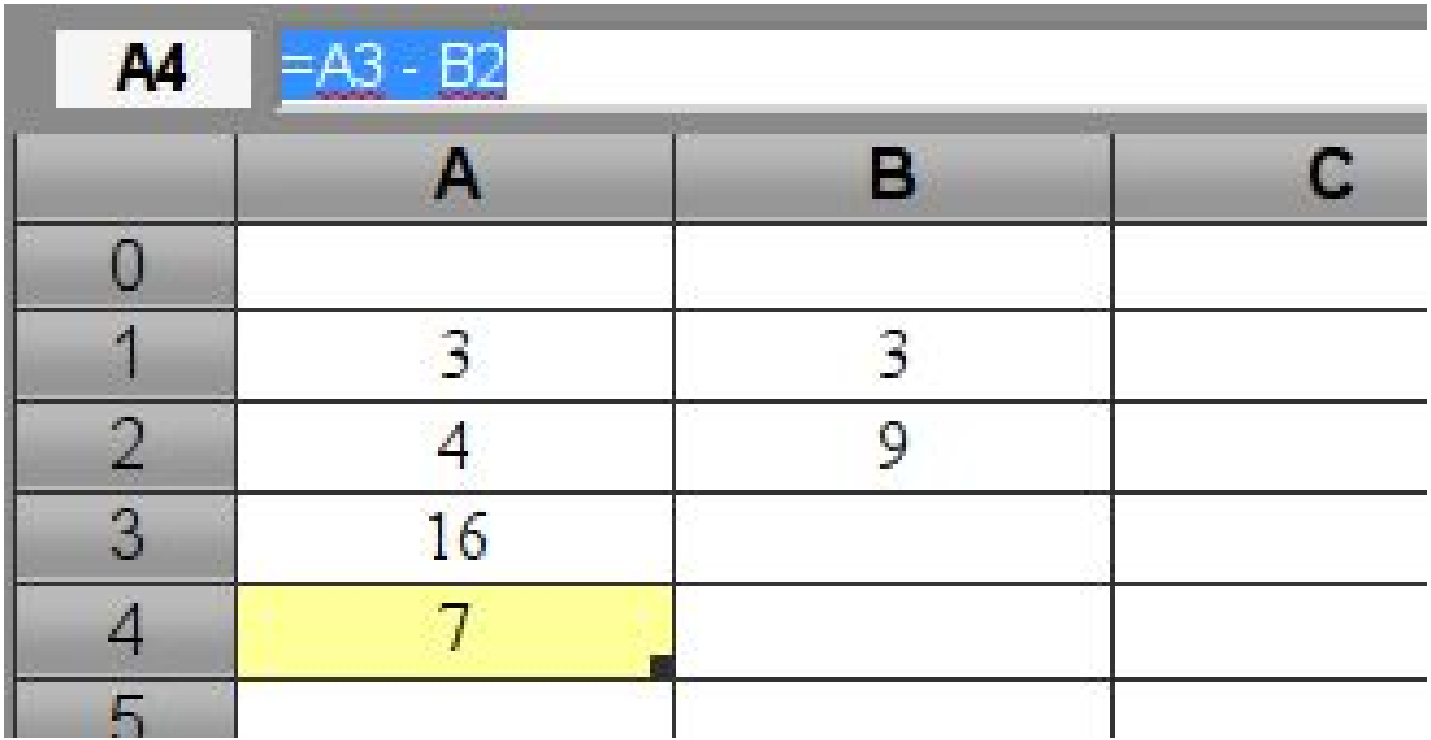
Calcul formel



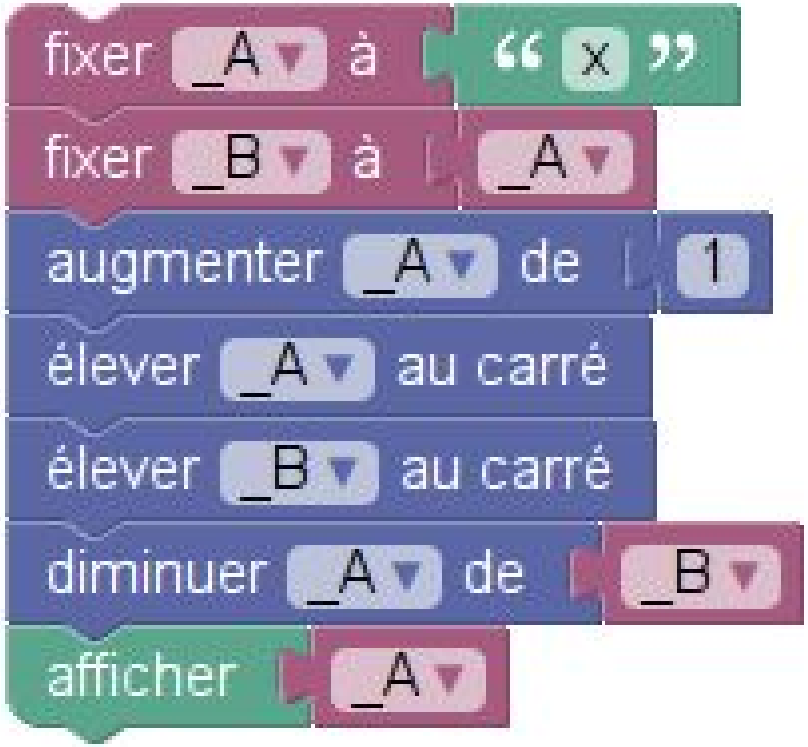
mathem@ALGO


fixer _A à 3
fixer _B à _A
augmenter _A de 1
élever _A au carré
élever _B au carré
diminuer _A de _B
afficher _A

mathem@ALGO



mathem@ALGO


fixer _A à “ x ”
fixer _B à _A
augmenter _A de 1
élever _A au carré
élever _B au carré
diminuer _A de _B
afficher _A

mathem@ALGO

	A	B	C
0			
1	x	x	
2	$x+1$	x^2	
3	$(x+1)^2$		
4	$(x+1)^2-x^2$		
5			
6			

mathem@ALGO

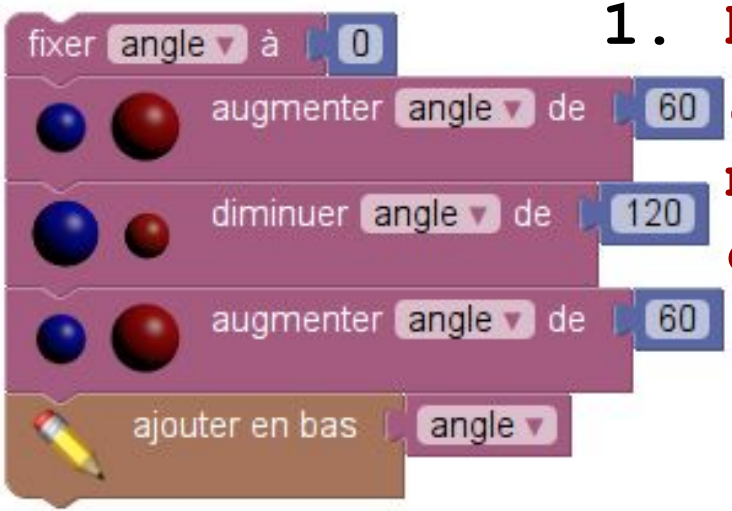
B4 =simplify(A4)

	A	B	C
0			
1	x	x	
2	$x+1$	x^2	
3	$(x+1)^2$		
4	$(x+1)^2-x^2$	$2x+1$	
5			

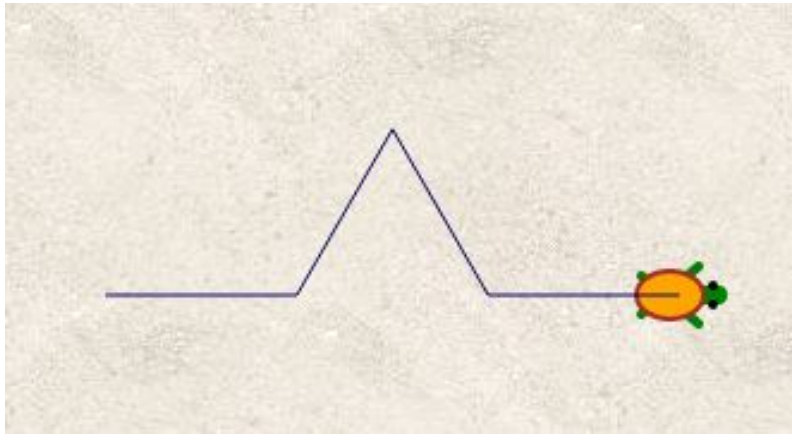
version graphique

1. **Montrer qu'à l'issue de ce script, la tortue a la même orientation qu'au départ.**
2. **Montrer qu'elle a alors parcouru 320 pixels.**
3. **Montrer qu'elle est alors à 240 pixels de son point de départ.**

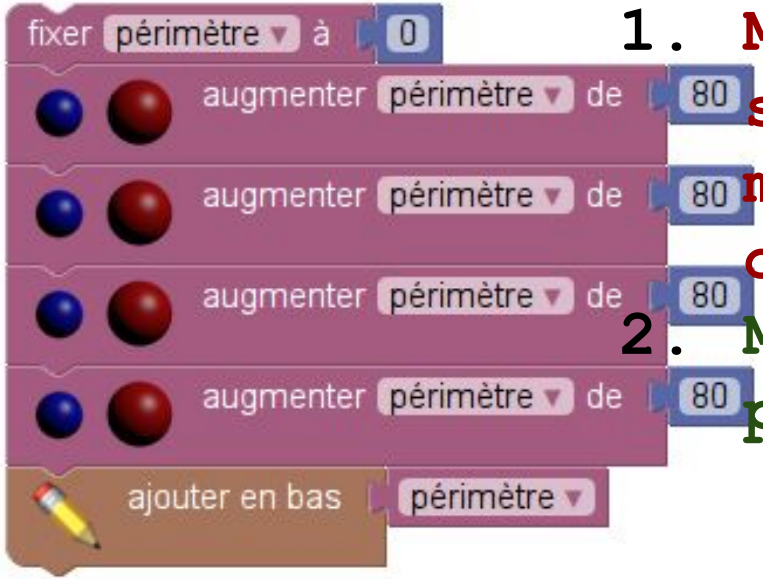
version graphique



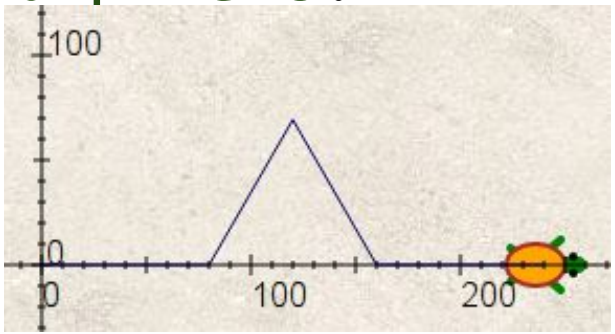
1. **Montrer qu'à l'issue de ce script, la tortue a la même orientation qu'au départ.**



version graphique



1. **Montrer qu'à l'issue de ce script, la tortue a la même orientation qu'au départ.**
2. **Montrer qu'elle a alors parcouru 320 pixels.**



version graphique

Montrer que ces deux programmes dessinent le même motif.