

Terminaison et correction

Fiche de synthèse

① DEUX QUESTIONS

Un algorithme **termine** s'il s'arrête toujours ; il est **correct** s'il donne le résultat attendu. On le prouve avec un variant et un invariant de boucle.

```
def division(a: int, b: int):
    """a et b entiers positifs non nuls.
    Renvoie le quotient et le reste
    de la division de a par b."""
    q = 0
    r = a
    while r >= b:
        q = q + 1
        r = r - b
    return (q, r)
```

② LA TRACE

La **trace** d'un algorithme note la valeur de chaque variable à chaque étape. Pour $a = 22$ et $b = 3$:

Tour	0	1	2	3	4	5	6	7
q	0	1	2	3	4	5	6	7
r	22	19	16	13	10	7	4	1

Après le septième tour, $r = 1 < 3$: la boucle s'arrête, et $22 = 3 \times 7 + 1$.

③ TERMINAISON : LE VARIANT

Définition 1 (Variant de boucle) Un **variant de boucle** est une quantité entière, positive ou nulle à l'entrée de chaque tour, qui **décroît strictement** à chaque tour. Un entier positif ne peut pas décroître indéfiniment : la boucle s'arrête.

Pour la division, r est un variant : à l'entrée d'un tour, $r \geq b > 0$; et r devient $r - b$, strictement plus petit car $b > 0$.

Une boucle `for i in range(n)` se termine toujours : son nombre de tours est fixé avant d'y entrer.

④ CORRECTION : L'INVARIANT

Définition 2 (Invariant de boucle) Un **invariant de boucle** est une propriété vraie avant le premier tour, et qui reste vraie après chaque tour.

Pour la division, l'invariant est $a = b \times q + r$ avec $r \geq 0$.

- **Initialisation** : avant la boucle, $q = 0$ et $r = a$, donc $b \times 0 + a = a$, et $r = a \geq 0$.
- **Conservation** : si $a = bq + r$ au début d'un tour, alors après le tour $b(q + 1) + (r - b) = bq + r = a$; et $r - b \geq 0$, puisque le tour n'a lieu que si $r \geq b$.
- **Conclusion** : à la sortie, l'invariant est vrai et la condition d'arrêt donne $r < b$. Donc $a = bq + r$ avec $0 \leq r < b$: c'est la division euclidienne.

☰ Prouver la correction d'une boucle.

1. Choisir un invariant.
2. Montrer qu'il est vrai avant la boucle, à l'aide des préconditions, les conditions que vérifient les entrées.
3. Montrer que, s'il est vrai au début d'un tour, il l'est encore à la fin.
4. À la sortie, combiner l'invariant et la négation de la condition du `while` pour obtenir la **postcondition**, la propriété voulue du résultat.

⑤ DEUXIÈME EXEMPLE

```
def multiplication(a, b):
    resultat = 0
    compteur = 0
    while compteur < b:
        resultat = resultat + a
        compteur = compteur + 1
    return resultat
```

Le variant est $b - \text{compteur}$, qui diminue de 1 à chaque tour. L'invariant est $\text{resultat} = a \times \text{compteur}$: vrai au départ, $0 = a \times 0$, et conservé, car $a \times \text{compteur} + a = a \times (\text{compteur} + 1)$. À la sortie, $\text{compteur} = b$, donc $\text{resultat} = a \times b$.

📅 À retenir.

- Un variant, entier positif strictement décroissant, prouve qu'une boucle se termine.
- Un invariant, vrai avant la boucle et conservé à chaque tour, prouve la correction.
- En sortie, invariant et condition d'arrêt donnent le résultat attendu.