

QCM 12 : savoir montrer qu'une itération (de type while) se termine

binôme du groupe 1 : Johann DOLIVET et Ivan MAUGIS

1 Énoncé de la question

Voici 4 algorithmes. Combien d'entre eux vont effectivement se terminer si on les exécute ?

- a) 0 b) 1 c) 2 d) 3

algorithme 1	algorithme 2	algorithme 3	algorithme 4
<pre> i = 0 j = 100 while j - i > 0: i = i - 1 j = j - 1 print(i,j) </pre>	<pre> i = 0 j = 100 while j - i > 0: i = i + 5 j = j + 4 print(i,j) </pre>	<pre> i = 0 j = 100 while j - i > 0: i = i + 1 j = j - 1 print(i,j) </pre>	<pre> i = 0 j = 100 while j - i > 0: i = i - 1 j = j + 1 print(i,j) </pre>

2 Correction, analyse

2.1 Correction

On illustre ici la notion de terminaison d'une boucle non bornée.

Le variant de boucle est la quantité $(j - i)$.

j est initialisé assez grand pour qu'une exécution manuelle soit fastidieuse. Ce qui ne l'empêche pas d'éclairer la situation. C'est une quantité entière, initialement positive, qui doit devenir négative ou nulle au bout d'un certain rang.

Soit

- $(j - i)'$ la valeur du variant en sorti de boucle,
- i' la valeur de i en sorti de boucle,
- j' la valeur de j en sorti de boucle.

Pour chaque algorithme $(j - i)$ est initialisé à $100 - 0 = 100$.

$(j - i)' = j' - i'$ $= (j - 1) - (i - 1)$ $= j - i - 1 + 1$ $(j - i)' = (j - i)$ $(j - i)$ étant constante, elle ne peut décroître vers 0.	$(j - i)' = j' - i'$ $= (j + 4) - (i + 5)$ $= j - i + 4 - 5$ $= (j - i) - 1$ $(j - i)' < (j - i)$ $(j - i)$ étant strictement décroissante, elle devient négative ou nulle au bout d'un certain temps.	$(j - i)' = j' - i'$ $= (j - 1) - (i + 1)$ $= j - i - 1 - 1$ $= (j - i) - 2$ $(j - i)' < (j - i)$ $(j - i)$ étant strictement décroissante, elle devient négative ou nulle au bout d'un certain temps.	$(j - i)' = j' - i'$ $= (j + 1) - (i - 1)$ $= j - i + 1 + 1$ $= (j - i) + 2$ $(j - i)' > (j - i)$ $(j - i)$ étant strictement croissante, elle ne peut décroître vers 0.
--	---	---	---

2.2 Objectif d'évaluation

On cherche à évaluer la capacité de l'élève à identifier le variant, la façon dont il évolue pour *in fine* aboutir à la preuve de terminaison (ou de non-terminaison).

Les erreurs attendues sont les suivantes :

- la non-identification du variant ;
- focalisation sur l'évolution de i et j ;
- confusion entre les indices i et j .

Choix des algorithmes :

- Dans l'algorithme 1, i et j décroissent tous les deux : l'élève peut se tromper en se disant « on va tomber dans les négatifs ».
- Dans les algorithmes 3 et 4, l'élève peut être déboussolé par des évolutions en sens contraire de i et j . Ce qui doit l'amener à prendre en compte le variant, qui lui a une évolution sans équivoque. Ces deux questions peuvent induire des confusions entre i et j .
- Dans l'algorithme 2, l'élève doit percevoir que la variation absolue de i et j ne compte pas, que seule la variation relative est importante ici.

3 Remédiation

Sous la forme d'une exercice guidé par un questionnement.

Pour chacune des boucles `while` situées dans les algorithmes suivants

- Identifier un variant de boucle.
- Identifier la condition d'arrêt de boucle sur ce variant.
- Identifier la façon dont ce variant évolue lors d'une itération.
- Conclure sur la terminaison ou non de la boucle.

Exemple :

<pre>i = 0 j = 100 while j - i > 0: i = i + 1 j = j - 1 print(i,j)</pre>	<p>a) Le variant est l'entier $(j - i)$.</p> <p>b) La boucle s'arrête dès que cet entier $(j - i)$ est négatif ou nul.</p> <p>c) Après un passage dans la boucle $(j-i)' = j' - i' = (j-1) - (i+1) = j-i-2 < (j-i)$.</p> <p>d) Le variant $(j-i)$ étant ainsi strictement décroissant d'un test à l'autre, il deviendra fatalement négatif à un moment. Donc la boucle s'arrête bel et bien.</p>
---	---

<pre>i = 0 j = 10 while j - i > 0: i = i + 5 j = j + 7 print(i,j)</pre>	<pre>i = 0 j = 10 while j - i > 0: i = i + 7 j = j + 5 print(i,j)</pre>	<pre>i = 0 j = 10 while j > i: i = i + 3 j = j + 2 print(i,j)</pre>	<pre>i = 0 j = 11 while j - i != 0: i = i + 2 j = j - 3 print(i,j)</pre>	<pre>T = [1,2,6,9,8,7] n = 0 while n < len(T): T.append(n) n = n + 1 print(T)</pre>
--	--	--	--	--

Commentaires sur le choix des algorithmes :

- Les quatre premiers algorithmes peuvent être exécutés intégralement de façon manuelle.
- Les deux premiers sont très similaires, mais un seul termine.
- Le variant du troisième est moins immédiat à identifier.
- La condition d'arrêt de boucle du quatrième est plus subtile.
- Le dernier oblige à s'interroger sur l'évolution de la taille d'une liste.