

Algo&Prog : Décompte des L – corrigé

Paul Gaborit

2014

Idée :

- Parcourir tous les caractères de la phrase.
- Incrémenter un compteur à chaque fois qu'on rencontre le caractère « L ».
- S'arrêter lorsqu'on rencontre un point (« . »).

Remarques :

- On reconnaît les signes d'un traitement par boucle :
 - le parcours d'une suite d'éléments,
 - un traitement à appliquer à chaque élément.
- Mais quelle structure de contrôle de boucle choisir ?

Lorsqu'on doit appliquer un traitement (ici, incrémenter un compteur si c'est un « L ») sur une suite d'éléments (ici la suite de caractères sur le ruban), on sait qu'on aura sûrement à utiliser une structure de contrôle de type boucle ou répétition.

Mais laquelle choisir parmi le Pour-FinPour, le TantQue-FinTantQue ou le Répéter-Jusqu'à ?

Voici quelques indications permettant de vous guider pour votre choix :

- Nombre d'éléments connus à l'avance :
→ structure Pour-FinPour
- Traitement spécifique du dernier élément :
→ structure TantQue-FinTantQue
- Reconnaissance du dernier élément après son traitement :
→ structure Répéter-Jusqu'à

Ces critères de choix ne sont ni exhaustifs ni même absolus. Mais généralement, cela donne un bon résultat...

Ici, nous ne connaissons pas à l'avance le nombre d'éléments à traiter (le nombre de caractère dans la phrase). Nous ne choisirons donc pas une structure de contrôle de type Pour-FinPour.

On peut reconnaître le dernier élément (le point) sans le traiter. D'ailleurs, on n'a pas à le traiter puisqu'on sait déjà qu'il ne peut pas être un « L ». Ce qui confirme que son traitement est spécifique (il n'y en a pas). Nous choisissons donc une structure de contrôle de type TantQue-FinTantQue.

COMPTER LES L SUR UNE PHRASE DE LA MACHINE À RUBAN()

Suppose que le ruban de la machine débute par une phrase finie par un point.

Variables : *NbL* (entier) nombre de «L».

CarCour (caractère) caractère courant.

```
Début
- . : - Initialisations
NbL ← 0
- . : - Lecture du premier caractère
Action : RAZ()
CarCour ← LIRECASE()
Tant Que (CarCour ≠ «.») Faire
- . : - Traitement du caractère courant
Si (CarCour = «L») Alors
| NbL ← NbL + 1
Fin Si
- . : - Lecture du caractère suivant
Action : AVANCER()
CarCour ← LIRECASE()
Fin Tant Que
- . : - Pas de traitement du dernier caractère
Action : AFFICHER(NbL)
Fin
```

Les commentaires dans l'algorithme donnent les lignes générales de l'utilisation d'une structure de contrôle de type TantQue-FinTantQue.

On y trouve (dans le désordre) :

Les initialisations Elle permettent de préparer le traitement à réaliser. Ici, on initialise notre compteur de « L » à zéro.

Tant Que (élément n'est pas le dernier) Faire C'est le test de continuation de notre boucle TantQue-FinTantQue. Il teste l'élément courant pour savoir si ce n'est pas le dernier. Ici, on teste si le caractère courant n'est pas un point.

La lecture du premier élément Avant de commencer la boucle, il faut commencer par lire le tout premier élément de notre suite. Il devient notre élément courant. Ici, il faut d'abord ramener la fenêtre de lecture au début du ruban puis lire le tout premier caractère.

Le traitement de l'élément courant C'est le traitement qui doit être appliquer à tous les éléments. Ici, on incrémente notre compteur si le caractère courant est un « L ».

Lecture de l'élément suivant C'est là qu'on passe d'un élément à un autre. Dans notre cas, cela consiste à faire avancer la fenêtre de lecture d'un cran puis à lire un nouveau caractère courant.

Le traitement du dernier élément C'est l'action qui a lieu après la boucle TantQue-FinTantQue et qui doit traiter le dernier élément, s'il y a lieu, puisqu'il n'a pas été traité dans la boucle. Ici, ce traitement est inutile (d'où le commentaire).

En essayant de retrouver ce schéma à chaque fois qu'on utilise une structure de contrôle de type TantQue-FinTantQue, cela permet de ne rien oublier et facilite à la fois la compréhension et la preuve de l'algorithme général.

On trouve une dernière instruction à la fin de l'algorithme pour afficher le décompte final des « L ».

Pour se convaincre du bon fonctionnement d'un tel algorithme, il faut pouvoir le dérouler *à la main* sur un ou plusieurs exemples.