

Quelques éléments de pseudo-code

L'indentation

L'obligation d'indenter les blocs d'instruction imposée par `Python` est une bonne habitude à adopter, même lorsqu'elle est optionnelle.

Affectations et comparaison. Les opérateurs de comparaison standards peuvent être utilisés, tels que :

`== < > ≤ ≥ != ≠ ...`

Les deux derniers symboles représentent la même chose, à savoir la différence de deux variables. Pour l'affectation, $x = y$ ou $x \leftarrow y$ permettent de représenter assez clairement cette opération.

Conditionnelle et boucles. L'importance est de clairement délimiter les blocs d'instruction (dans le cadre des conditionnelles, les `Si/Sinon Si/Sinon`). Par exemple :

```
1 Si condition Alors
2 |   Instructions
3 Fin
4 Sinon si condition Alors
5 |   Instructions
6 Fin
7 Sinon
8 |   Si condition Alors
9 | |   Instructions
10 | Fin
11 | Sinon
12 | |   Instructions
13 | Fin
14 Fin
```

```
1 Si condition Alors
2 |   Instructions
3 Sinon si condition Alors
4 |   Instructions
5 Sinon
6 |   Si condition Alors
7 | |   Instructions
8 | Sinon
9 | |   Instructions
```

Les boucles **Tant que** et **Pour** peuvent être gérées de manière relativement similaire (l'écriture plus condensée proposée pour les conditionnelles est bien sûr également possible) :

```
1 Pour i de 1 à n Faire
2 |   Instructions
3 Fin
```

```
1 Pour e dans Ensemble
   Faire
2 |   Instructions
3 Fin
```

```
1 Tant que condition Faire
2 |   Instructions
3 Fin
```

Les entrées-sorties. Pour simuler des utilisations des fonctions `print` et `input` on peut imaginer :

```
1 variable ← lire()
```

```
1 Afficher([liste des arguments])
```

Les tableaux. Voici quelques éléments pour la manipulation des tableaux :

- par souci de simplicité pour la manipulation et la preuve de correction et de terminaison des algorithmes, **les tableaux seront indicés de 1 au nombre d'éléments qu'ils contiennent**
- la taille d'un tableau `t` est accessible par `taille(t)`
- un élément du tableau `t` est noté `t[i]` si le tableau est à une dimension, `t[i][j]` s'il est à deux dimensions.

Les fonctions. Plusieurs niveaux de détails sont possibles pour les fonctions. Les éléments mis entre `[]` sont facultatifs mais peuvent faciliter la lecture de la fonction. Les accolades sont utilisées pour délimiter le début et la fin de la fonction dans de nombreux langages de programmation.

```
1 Fonction (arguments [type]) [retourne type]{
2 |   Instructions
3 |   Retourner resultat
4 }
```