

Les bases de l'algorithmique

Objectifs :

- Connaître et écrire des instructions conditionnelles (si alors) et des répétitives (pour, tant que)
- Ecrire un algorithme en pseudo-code
- Faire fonctionner un algorithme à la main
- Compter le nombre d'opérations dans un algorithme (introduction au coût d'un algorithme)
- Compter le nombre d'itérations dans un algorithme (introduction au coût d'un algorithme)

Variables et affectations

Dans un algorithme, des variables sont utilisées pour stocker des valeurs qui peuvent être des nombres (entiers, réels, ...), des chaînes de caractères, des booléens et beaucoup d'autres choses encore.

Le symbole \leftarrow est celui de l'affectation. Par exemple, $a \leftarrow 2$ signifie « la variable a prend la valeur 2 ».

L'opération arithmétique div est le quotient de la division entière et mod est le reste de la division entière.

Par exemple, $14 \text{ div } 5 = 2$ et $14 \text{ mod } 5 = 4$ car $14 = 5*2+4$, rappelez-vous :

$$\begin{array}{r} 14 \quad | \quad 5 \\ \underline{4} \quad | \\ 4 \quad | \quad 2 \end{array}$$

Dans toute la suite du document, toutes les variables sont des entiers ou des booléens.

Exercice 1 :

$N \leftarrow 2$
$P \leftarrow 3$
$N \leftarrow P + 1$
$P \leftarrow N$

Que valent N et P après l'exécution de cet algorithme ?

Exercice 2 :

$A \leftarrow 8$
$B \leftarrow (A \geq 10)$

Que vaut B après l'exécution de cet algorithme ?

Exercice 3 :

$P \leftarrow N \text{ mod } 5$
$P \leftarrow N \text{ div } P$

Supposons que N vaut 2 ; que vaut P après l'exécution de cet algorithme ?

Exercice 4 :

Pour chacun des trois algorithmes précédents, dénombrer le nombre d'affectations, le nombre d'opérations arithmétiques et le nombre de comparaisons.

Instructions conditionnelles

Une instruction conditionnelle, ou instruction de test, permet de faire des choix dans un algorithme en fonction de la valeur d'une condition. On parle souvent d'une instruction **si-alors**.

Une condition est une expression qui est soit vraie, soit fausse, c'est donc un booléen.

Exercice 5 :

```
si A ≥ 0
    alors A ← A - 1
    sinon A ← A + 1
fin si
```

Supposons que A vaut 1 au début ; que vaut-il après l'exécution de cet algorithme ?

Exercice 6 :

```
si A ≥ 0
    alors A ← A - 1
fin si
si A < 0
    alors A ← A + 1
finsi
```

L'algorithme ci-contre (au sens où une même valeur de A initiale induit une même valeur de A après leurs exécutions) à celui présenté dans l'exercice 5 ?

Exercice 7 :

Pour chacun des deux algorithmes précédents, dénombrer le nombre d'affectations, le nombre d'opérations arithmétiques et le nombre de comparaisons.

Exercice 8 :

Un site internet de développement de photos propose le tirage sur papier des photos au tarif de 0,11 € l'unité ; le tarif passe à 0,08 € l'unité pour une commande d'au moins 200 photos. Proposez un algorithme qui affiche le montant à payer selon le nombre de photos à développer.

Répétitives énumérées (boucles Pour)

La boucle Pour est une structure qui permet de répéter plusieurs fois l'exécution d'une ou plusieurs instructions. On parle de répétitive énumérée car on peut énumérer à l'avance les valeurs de l'itérateur de boucle.

Par exemple : dans la boucle « pour i de 0 à 3 » ; on peut énumérer les valeurs de l'itérateur i qui prendra les valeurs 0, 1, 2 puis 3.

Ainsi, une boucle Pour s'utilise lorsque l'on connaît à l'avance le nombre d'itération à effectuer.

Chaque passage dans la boucle Pour s'appelle une itération.

Exercice 9 :

```
pour i de 2 à 5 faire
    A ← A + i
fin pour
```

Supposons que A vaut initialement 10 ; combien vaut-il après l'exécution de cet algorithme ?

Exercice 10 :

```

pour I de 1 à 4 par pas de 1 faire
    si I est pair
        alors V ← V + I*I
    fin si
fin pour

```

Supposons que V vaut 0 initialement ; combien vaut-il après l'exécution de cet algorithme ?

Exercice 11 :

On considère l'algorithme précédent.

1. Combien de fois la condition I est pair est-elle évaluée ?
2. Combien de fois la variable V change-t-elle de valeur ?
3. Dénombrez alors l'ensemble des opérations : opérations arithmétiques, comparaisons, affectations.

Exercice 12 :

```

pour i de 0 à 3 faire
    pour j de 0 à 2 faire
        A ← A + i*j
    fin pour
fin pour

```

Supposons que A vaut 1 initialement ; combien vaut-il après l'exécution de cet algorithme ?

Répétitives conditionnées (boucles Tant que)

La boucle Tant que est aussi une structure qui permet de répéter plusieurs fois l'exécution d'une ou plusieurs instructions. On parle de répétitive conditionnée car c'est une condition qui va déterminer le nombre d'itérations à effectuer.

Ainsi, une boucle Tant que s'utilise lorsque le nombre d'itérations n'est pas connu à l'avance. Chaque passage dans la boucle Tant que s'appelle aussi une itération.

Exercice 13 :

```

tant que D > 3 faire
    D ← D - 3
fin tant que

```

Supposons que D vaut 7 initialement ; que vaut-il après l'exécution de cet algorithme ?

Exercice 14 :

On considère l'algorithme précédent.

1. Supposons que D vaut 2 après l'exécution de cet algorithme. Quelles valeurs parmi celles proposées ci-dessous pouvait-il avoir initialement ? -2 ; 2 ; 6 ; 14 ; 25.
2. Supposons que D vaut 25 initialement ; combien d'itérations (c'est-à-dire de passages dans la répétitive) ont lieu avant que l'exécution de cet algorithme ne se termine ?

Exercice 15 :

Proposez un algorithme qui utilise une boucle Tant que qui a pour but d'afficher le quotient de la division entière d'un entier naturel A par un entier naturel B.

Les fonctions

Voyons comment écrire l'algorithme en pseudo-code d'une fonction. Prenons l'exemple d'une fonction nommée `maxi` qui prend deux nombres en paramètres et qui renvoie le maximum des deux. On l'écrit ainsi en pseudo-code :

```
fonction maxi(a,b)
  si a > b
    alors retourner a
  sinon retourner b
finsi
```

Exercice 16 :

Proposez le pseudo-code d'une fonction `mini` qui renvoie le minimum des deux nombres donnés en paramètres.

Exercice 17 :

Proposez le pseudo-code d'une fonction `mini4` qui renvoie le minimum des 4 nombres donnés en paramètres. Contrainte : vous devez faire appel à la fonction `mini` !

Exercice 18 :

Les élèves d'une classe ont eu 4 notes. Le professeur, très généreux, décide de retirer la note la plus basse avant de calculer la moyenne de chaque élève.

Proposez le pseudo code d'une fonction `nouvelleMoyenne` qui prend les 4 notes en paramètres et qui renvoie la moyenne calculée après avoir retiré la moins bonne.

Source :

- Cours d'algorithmique, DIU EIL, Nantes