Définition. Un *algorithme* est une suite finie d'opérations élémentaires permettant de résoudre un problème donné.

Exemple. …

Trois parties (qui peuvent être mélangées) composent un algorithme :

— entrée des données ;

— traitement des données ;

— sortie des résultats.

1 Variables

Définition. Une *variable* est un espace en mémoire, portant un nom, dans lequel on peut stocker une valeur.

L'instruction `b prend la valeur a` signifie : aller chercher en mémoire la valeur de `a`, et placer cette valeur dans `b`. D'autres notations possibles sont : `b` ← `a` ; `Affecter a à b` ; `a` → `b` ; `a = b` ; etc.

Exemple.

Instructions	x	y
Début		
`x prend la valeur 3`		
`y prend la valeur 3x`^{2} `+ 2x`		
`x prend la valeur 2y`		

2 Entrées / Sorties

Il est possible de demander à l'utilisateur d'affecter une valeur à une variable, avec une instruction du type `Lire a`. Le programme attend

que l'utilisateur saisisse une valeur au clavier, puis place dans l'espace mémoire correspondant à `a` cette valeur.

Il est possible d'afficher quelque chose à l'écran avec l'instruction `Afficher a`.

Exemple. Que fait l'algorithme suivant ?

```
Lire a
a ← a × a
Afficher a
```

3 Conditionnelles

Définition. Pour résoudre certains problèmes, il est parfois nécessaire de faire un test pour savoir si on doit exécuter une tâche ou une autre. *Si* le test est vrai, *alors* on exécute une tâche, *sinon* on exécute une autre tâche.

Exemple.

```
Afficher "Longueur des trois cotes."
Lire A
Lire B
Lire C
Si A = B et B = C
Alors
  Afficher "Le triangle est equilateral."
Sinon
  Si A = B ou B = C ou A = C
  Alors
    Afficher "Le triangle est isocele."
  Sinon
    Afficher "Le triangle est scalene."
  FinSi
FinSi
```

4 Boucles

Certains problèmes nécessitent de répeter un ensemble d'instructions plusieurs fois.

Il existe deux types de boucles :

— les boucles `Pour i allant de 1 à n` exécutent `n` fois la boucle, pour chacune des valeurs possibles de `i` ;

— les boucles `While condition` exécutent la boucle tant que la condition n'est pas remplie.

Exemple. Affichage de la table de multiplication de 9.

```
Pour i allant de 1 a n
Faire
   Afficher 9×i
FinPour
```

Exemple. Calcul de l'entier n tel que la somme des entiers de 1 à n fait 2016.

```
0 → somme
0 → n
Tant que somme < 2016
Faire
   n + 1 → n
   somme + n → n
FinTantque
Si somme = 2016
Alors
   Afficher n
Sinon
   Afficher "Impossible"
FinSi
```

5 Exercices

Exercice 1. Un magasin offre 5% de réduction sur le montant total d'un achat si celui-ci est supérieur à 100 €. Écrire un algorithme qui lit en entrée le montant total, et affiche le montant après l'éventuelle réduction.

Exercice 2. Une bactérie double sa population chaque jour. Écrire un algorithme qui calcule la population de bactéries après trente jours, à partir d'une seule bactérie au départ.

Exercice 3. Un enfant veut acheter un jeu à 245 €. Ses parents lui donnent un euro la première semaine, deux euros la deuxième semaine, et ainsi de suite. Écrire un algorithme qui calcule à partir de combien de semaine l'enfant va-t-il pouvoir acheter son jeu ?

Exercice 4. On appelle *parfait* un nombre qui est égal à la somme de ses diviseurs, sauf lui même ($6 = 1 + 2 + 3$ est parfait ; $8 \neq 4 + 2 + 1$ n'est pas parfait). Écrire un algorithme qui vérifie si un nombre est parfait. Écrire un second algorithme qui cherche tous les nombres parfaits inférieurs à un nombre donné.