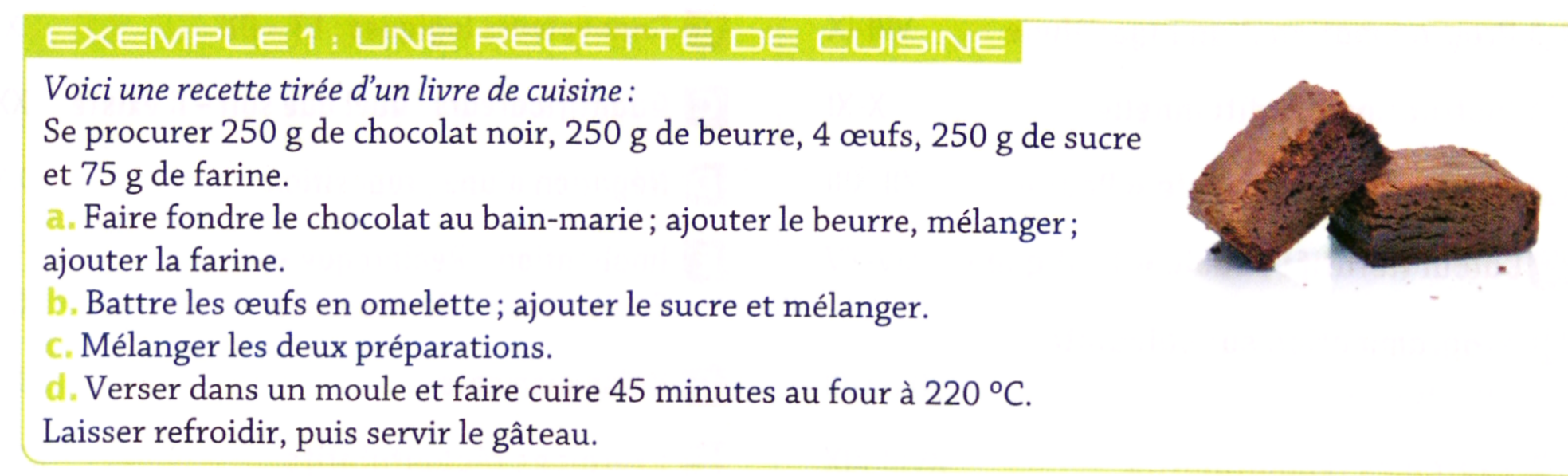
**Notion d’algorithmique**

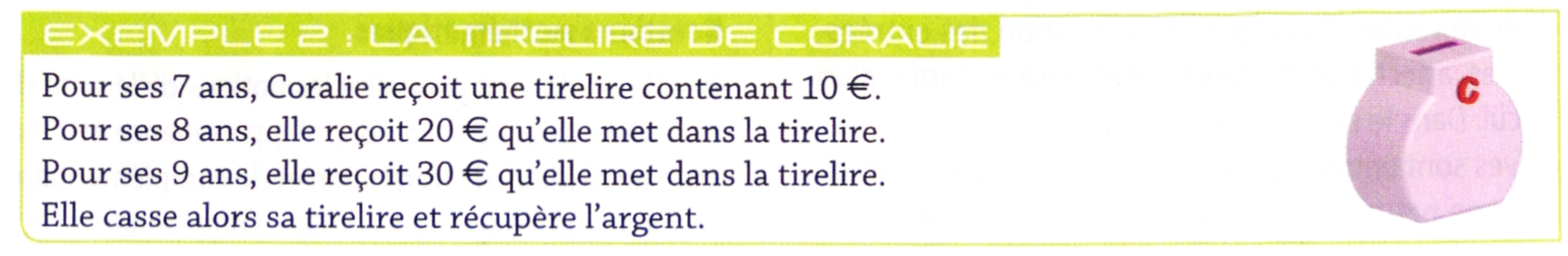
**I/ Des exemples d’algorithmes au quotidien**

*Sans le savoir, on utilise quotidiennement des algorithmes comme la recette de cuisine ci-dessous.*

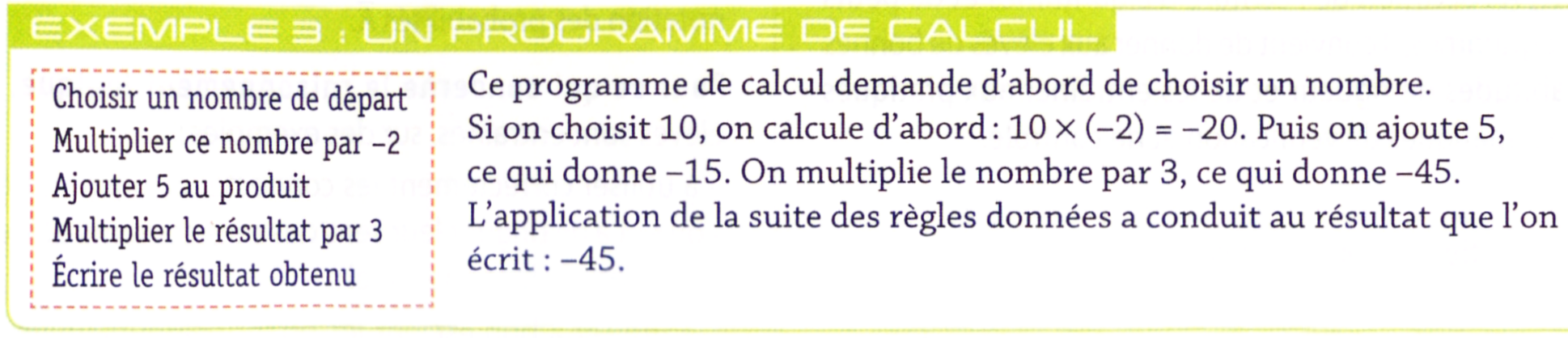
******

Ce texte décrit les opérations à réaliser successivement pour faire un moelleux au chocolat.

**A partir** des ingrédients de la recette, avec les quantités requises, le texte donne les **règles** à suivre : il s’agit des **étapes** a, b, c et d qui **s’enchaînent**. Le **résultat** est le gâteau fini.



**Au départ**, Coralie a une tirelire garnie. Après **deux étapes** au cours desquelles le **contenu** de la tirelire **se modifie**, elle découvre le **résultat** en cassant sa tirelire.



**Algorithme** : ...............................................................................................................................................

....................................................................................................................................................................

A noter : Le mot « algorithme » vient du nom mathématicien persan **Al-Khwarizmi** (780 – 850). il a écrit en langue arabe le plus ancien traité d’algèbre dans lequel il décrivait des procédures de résolution pas à pas d’équations.

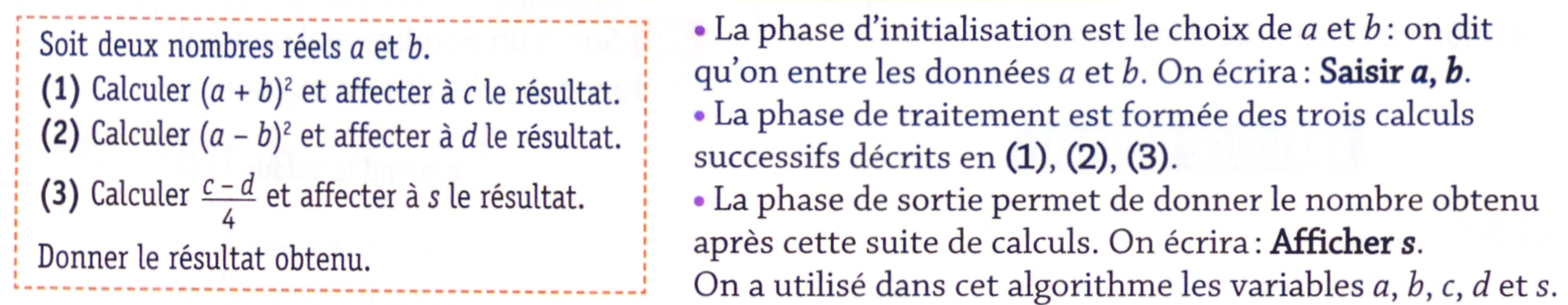
**II/ Ecriture d’un algorithme**

**1/ Structure d’un algorithme**

Un algorithme comprend :

* une phase **d’initialisation** : on ***déclare*** et ***initialise*** les variables et on ***entre*** les données ;
* une phase de **traitement** du problème ;
* une phase de **sortie** des résultats.

**Exemple** :



**2/ Formalisme d’écriture d’un algorithme**

|  |  |
| --- | --- |
| *Ecriture d’un algorithme en langage naturel* | *En langage formel* |

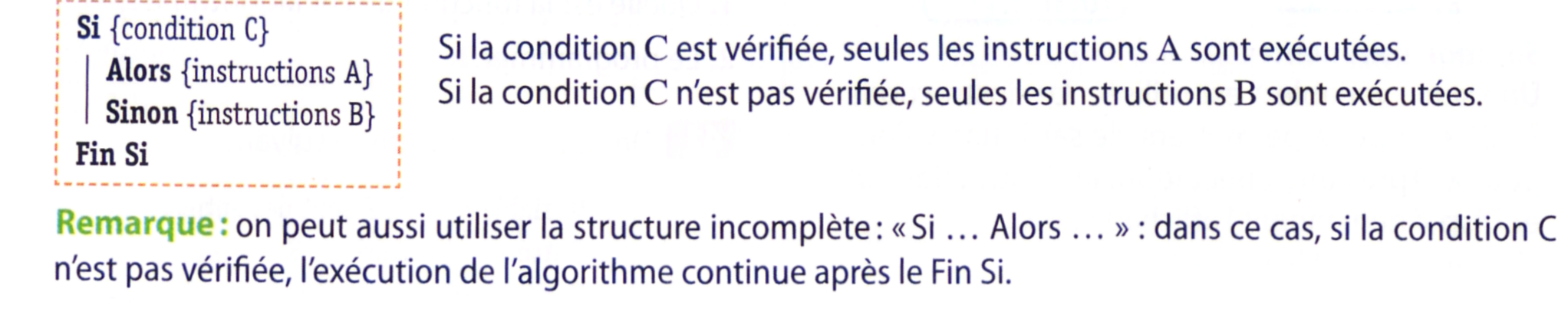
**Exemple** :

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 1/ Quelles sont les **variables** ?  ..........................................................................  2/ **Déterminer** dans l’algorithme la (ou les) ligne(s) qui correspondent aux étapes :  les entrées : ........................................... ;  le traitement : ........................................ ;  la sortie : ................................................ . | |
|  | |  | |
| **Exercice 1 :**    ...............................................................................  ............................................................................... | **Exercice 2 :** | |
| **Exercice 3 :**  **Ecrire** un algorithme affichant la somme S et le produit P de deux entiers a et b donnés.  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ............................................................................... | **Exercice 4 :**  **Ecrire** un algorithme qui demande une température C (exprimée en degrés Celsius), puis la transforme en degrés Fahrenheit F, sachant que l’on a la relation F = 1,8 x C + 32.  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ...............................................................................  ............................................................................... | |

**III/ Instruction conditionnelle**

Une **instruction conditionnelle** permet d’exécuter **une partie** d’un algorithme en fonction d’une **condition** (vérifiée ou non) fixée par le programmeur.

Voici la structure :



**Exemple :**

|  |  |
| --- | --- |
|  | 1/ Pour chacune des entrées suivantes, **déterminer** la valeur affichée par l’algorithme de sortie :   1. A = 4 et B = 7 : ......................................... 2. A = 12 et B = 9,2 : ....................................   2/ Que fait cet **algorithme** ?  .................................................................................  ................................................................................. |

|  |  |
| --- | --- |
| **Exercice 1 :** | 1/ Quel est le prix **payé** par un client effectuant :   1. 28 photocopies ? ....................................... 2. 30 photocopies ? ....................................... 3. 52 photocopies ? .......................................   2/ **Déterminer** le prix unitaire des 30 premières photocopies et celui des photocopies suivantes.  ..............................................................................  ..............................................................................  3/ La commerçante décide de changer ses tarifs : les 20 premières photocopies seront facturées  0,25 euros et les suivantes 0,10 euros. **Modifier** l’algorithme. |

|  |  |
| --- | --- |
| **Exercice 2 :** | 1/ Pour chacune des entrées suivantes, **déterminer** la valeur affichée en sortie :   1. A = 5 et B = 9 : ................................. 2. A = 2 et B = -2 : ................................ 3. A = -3 et B = -7 : ............................... 4. A = 8 et B = 2 : .................................   2/ Que fait cet **algorithme** ?  ............................................................................  ............................................................................  ............................................................................ |
| **Exercice 3 :**  **Ecrire** un algorithme permettant de calculer le prix à payer pour un utilisateur de téléphone portable lorsque celui-ci bénéficie d’une forfait de 2 heures pour 8 euros et où chaque minute au-delà du forfait est facturée 0,20 euro. | ............................................................................  ............................................................................  ............................................................................  ............................................................................  ............................................................................  ............................................................................  ............................................................................  ............................................................................  ............................................................................  ............................................................................  ............................................................................ |

**IV/ Instruction itérative : boucle « POUR »**

Une **boucle « POUR »** est utilisée lorsque l’on veut **recommencer plusieurs fois** un même **bloc d’instructions**. Cette boucle est finie.

Voici sa structure :

|  |  |
| --- | --- |
| On peut donner n’importe quel nom à la variable.  A chaque tour de la boucle, **sa valeur augmente constamment de 1**, on appelle cela un « pas » |  |

**Exemple :**

|  |  |
| --- | --- |
|  | Que permet d’obtenir **l’algorithme** suivant ?  ...................................................................................  ................................................................................... |

|  |  |
| --- | --- |
| **Exercice 1 :** | .................................................................................  .................................................................................  ................................................................................. |

|  |  |
| --- | --- |
| **Exercice 2 :**     1. ................... et .................... 2. ................... et .................... | **Exercice 3 :**    On trouve ......................................... |

|  |  |
| --- | --- |
| **Exercice 4 :**  **Ecrire** un algorithme qui permette d’obtenir la somme des *n* premiers carrés 1² + 2² + 3² + ... + n² | .................................................................................  .................................................................................  .................................................................................  .................................................................................  .................................................................................  .................................................................................  .................................................................................  ................................................................................. |

**V/ Instruction itérative : boucle « TANT QUE »**

La **boucle « TANT QUE »** est une structure itérative avec **fin de boucle conditionnelle. Elle est utilisée quand on ne sait pas à l’avance combien d’itérations il y aura**.

Voici sa structure :

|  |  |
| --- | --- |
| Remarque : La condition C est testée en début de boucle, donc, si elle n’est pas vérifiée au départ, la boucle n’est jamais effectuée. |  |

**Exemple :**

|  |  |
| --- | --- |
| Voici un algorithme : | **Compléter** l’algorithme pour qu’il affiche tous les multiples entiers naturels de l’entier A strictement inférieurs à 1000.  ................................................................................  ................................................................................ |

|  |  |
| --- | --- |
| **Exercice 1 :**    ........................................................................... | **Exercice 2 :** |