Chapitre 0 - Découverte

0.1 Vers l'algorithmique

Les problèmes que l'on rencontre habituellement en mathématiques trouvent en général leurs solutions dans le calcul à la main et le raisonnement. Il existe cependant toute une catégorie de problèmes pour lesquels ces techniques demeurent lourdes et inefficaces.
Pour comprendre, considérons les cinq problèmes suivants.

5 problèmes …

- **Problème n°1** : Calculer A = 2 + 5 × 6 .
- **Problème n°2** : Calculer B = 25,68 × 14,235 + 123,184.
- **Problème n°3** : Trouver le nombre x tel que $5x + 1 = 7$.
- **Problème n°4** : Additionner tous les nombres entiers de trois chiffres.
- **Problème n°5** : Quel est le plus petit nombre entier n tel que $n^3 + 3n > 100\,000$?

Résolutions

✓ **Problème n°1** : On calcule aisément A, à la main.

$$\begin{aligned} A &= 2 + 5 \times 6 \\ &= 2 + 30 \quad \text{(la multiplication est prioritaire sur l'addition)} \\ &= 32 \end{aligned}$$

✓ **Problème n°2** : Le calcul de B nécessite une calculatrice pour ne pas perdre trop de temps. On obtient alors B = 488,7388.

✓ **Problème n°3** : Pour déterminer x, on soustrait d'abord 1 de chacun des membres de l'égalité puis on les divise par 5.

$$\begin{aligned} &5x + 1 = 7 \\ &5x + 1 - 1 = 7 - 1 \qquad (*) \\ &5x = 6 \\ &\frac{5x}{5} = \frac{6}{5} \qquad (**) \\ &x = \frac{6}{5} = 1{,}2 \end{aligned}$$

Avec l'habitude on peut se passer d'écrire les égalités (*) et (**).

Les calculs à la main ou à la calculatrice et l'utilisation des règles algébriques de base nous ont permis de résoudre les trois premiers problèmes. Concernant la résolution des deux autres, l'usage de ces méthodes serait long, laborieux et source d'erreurs. Alors, comment les résoudre ?

✓ **Problème n°4** : Les nombres entiers de trois chiffres sont 100, 101, 102, … , 997, 998, 999. Les additionner soi-même un à un prendrait trop de temps. Cependant, on peut indiquer la démarche à suivre à un ordinateur qui effectuerait alors lui-même les calculs quasi-instantanément.

Concernant la méthode, on pourrait lui demander d'effectuer les actions :

► Partir de 0.
► Puis, ajouter 100, puis 101, puis 102, puis 103, etc … jusqu'à 999.
► Enfin, afficher le résultat.

Nous verrons dans le chapitre 1 comment traduire et faire exécuter ces actions à un ordinateur. Nous nous contenterons pour l'instant de dire qu'on obtiendra le résultat 494 550.

*Une série d'actions élémentaires, assemblées de manière logique, comme les précédentes, est appelée **algorithme**.*

*Un algorithme a un **rôle**, c'est-à-dire un but. Le rôle de celui que nous venons d'écrire est le calcul de la somme des nombres entiers de trois chiffres.*
*Pour être interprété par un ordinateur, un algorithme devra être traduit au préalable dans un langage qu'il comprend, appelé **langage de programmation**. Il en existe des centaines, répondant chacun à des objectifs particuliers. Les plus connus sont Python, C, C++, C#, Java et Php.*

Dans ce cours, nous avons choisi de travailler avec le langage Scratch car c'est celui qui convient le mieux à l'apprentissage des notions abordées au collège.

✓ **Problème n°5** : On peut observer que $n^3 + 3n > 100\,000$ n'est pas vrai pour des petites valeurs de n.

Par exemple, l'inégalité $n^3 + 3n > 100\,000$ est fausse pour $n = 0$:

$0^3 + 3 \times 0 = 0$ et on n'a pas $0 > 100\,000$.

De même, elle est fausse pour $n = 1$ et $n = 2$:

- $1^3 + 3 \times 1 = 1 + 3 = 4$ et on n'a pas $4 > 100\,000$.
- $2^3 + 3 \times 2 = 8 + 6 = 14$ et on n'a pas $14 > 100\,000$.

Cependant, pour de grandes valeurs de n, l'inégalité $n^3 + 3n > 100\,000$ est vraie. Par exemple pour $n = 100$:

$$100^3 + 3 \times 100 = 1\,000\,000 + 300 = 1\,000\,300$$

et on a bien $1\,000\,300 > 100\,000$.

De même que pour le problème n°4, il serait long et fastidieux de déterminer à la main la valeur de n cherchée. On peut alors écrire un algorithme :

► Partir de $n = 1$.
► Puis, tant que $n^3 + 3n$ ne dépasse pas 100 000, augmenter n de 1.
► Enfin, afficher la valeur de n obtenue.

Après avoir traduit cet algorithme dans un langage de programmation, on obtiendrait après exécution la valeur $n = 47$.

Lorsqu'on aborde un problème que l'on veut résoudre par une solution algorithmique, on doit d'abord le découper en une suite d'actions élémentaires donnant naissance à un algorithme.

Dès que les problèmes se complexifient, les algorithmes deviennent de plus en plus longs. Il est donc important qu'ils respectent un certain nombre de règles de présentation. Même s'il est encore un peu tôt pour en comprendre le fonctionnement et la syntaxe, voici ci-dessous les algorithmes des problèmes 4 et 5 sous une forme conventionnelle.

▪ Algorithme pour résoudre le problème 4 :

```
Variables : s,k (nombres)
Début
|s prend la valeur 0
|Pour k de 100 à 999 Faire
||s prend la valeur s+k
|FinPour
|Afficher s
Fin
```

▪ Algorithme pour résoudre le problème 5 :

```
Variable : n (nombre)
Début
|n prend la valeur 1
|TantQue n^3+3*n ≤ 100000 Faire
||n prend la valeur n+1
|FinTantQue
|Afficher n
Fin
```

*Lorsqu'on traduit un algorithme dans un langage de programmation, on dit qu'on **l'implémente**. Cette traduction est appelée **programme informatique**.*
Même s'il est aussi un peu tôt pour comprendre ce qui suit (figures 0.1 et 0.2), il s'agit de l'implémentation en Scratch des deux algorithmes précédents.

Figure 0.1

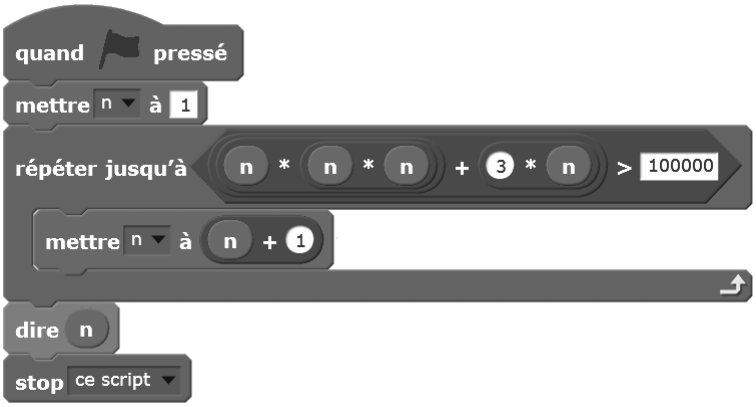

Figure 0.2

0.2 Présentation de Scratch

Scratch est un logiciel libre, conçu dans le but de s'initier dès le plus jeune âge à l'algorithmique et à la programmation informatique.
Dans cet ouvrage, nous utilisons la version 2.0 de Scratch, qui est la plus récente au moment où il est écrit.

Scratch est ludique et pratique pour apprendre sans difficulté de formalisme, le codage informatique et les bonnes pratiques algorithmiques.
En effet, il n'est nul besoin de maîtriser une syntaxe d'écriture comme c'est traditionnellement le cas pour les langages de programmation usuels. Dans le cas de Scratch, il suffit de glisser, déposer et assembler des blocs pour définir les instructions à effectuer. Nous découvrirons plus en détail ce fonctionnement dans la prochaine partie.

Pour télécharger la version offline de Sratch 2.0 :

- Se rendre à l'adresse : https://scratch.mit.edu/scratch2download
- Sélectionner son système d'exploitation (Mac OS, Windows ou Linux).
- Puis démarrer le téléchargement.

Figure 0.3

Après téléchargement puis installation, lancer Scratch.

On arrive alors à l'écran ci-après (figure 0.4) où on peut distinguer :

- **Le menu des blocs d'instruction**. Ces blocs sont des sortes de briques que l'on assemble pour créer des programmes. Ils sont classés par catégories.
Les blocs « Mouvement » gèrent le mouvement des objets, les blocs « Apparence » gèrent leur apparence, etc …

▪ **La zone des scripts**. C'est le lieu où on dépose et assemble les blocs par glisser-déposer.

▪ **La scène**. C'est ici que ce déroule l'exécution du programme.

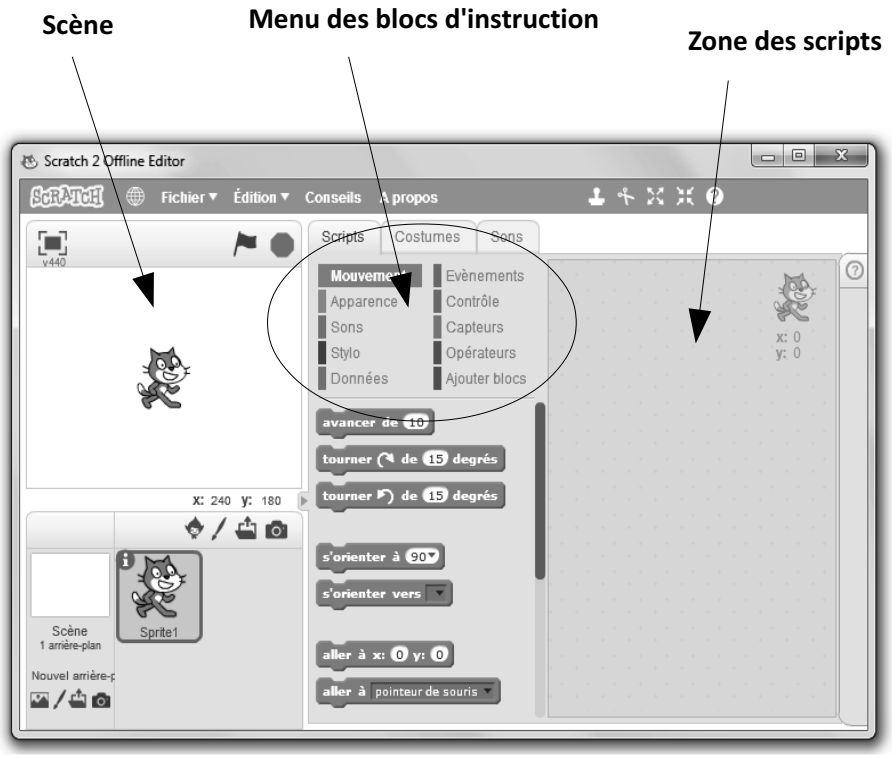

Figure 0.4