

**TD 1: Notion d'algorithme**

Les exercices non marqués d'un ♣ sont considérés comme acquis d'une semaine sur l'autre.

**Exercice : Modélisation d'un problème et notion d'algorithme**

Un passeur doit faire passer de l'autre côté d'une rivière un loup, une chèvre et un chou. Pour cela, il ne peut transporter qu'un seul d'entre eux à la fois dans sa barque. On ne sait pas trop ce qui arriverait si le loup venait à rester seul en présence de la chèvre, ou si la chèvre se retrouvait seule en présence du chou.

1. Décrivez une suite d'instructions simples (un premier algorithme!) à effectuer scrupuleusement par le passeur garantissant que tout le monde traverse la rivière sain et sauf. Les instructions disponibles sont :
  - `charge(objet)` (exemple: `charge(chèvre)`)
  - `décharge(objet)`
  - `traverse()`

**Indication :** Ne cherchez pas à résoudre le problème de tête; au contraire, l'objectif est d'apprendre à raisonner sur papier pour aborder ensuite des problèmes plus complexes et pouvoir transmettre ce raisonnement. Cherchez une manière de représenter l'état du système, et décrivez sur votre feuille comment celui-ci évolue au fur et à mesure des instructions.

2. Identifiez une séquence d'instructions qui se répète et proposer une nouvelle instruction les combinant – d'un niveau d'abstraction plus élevé – permettant de simplifier l'écriture de l'algorithme précédent.

**Exercice : Modélisation d'un problème et notion d'algorithme**

Soient deux cruches de capacités respectives 5 et 7 litres. Ces cruches ne sont pas graduées. Le but de ce problème est que l'une des deux cruches contienne 4 litres. Vous avez accès à un robinet et à une évacuation d'eau, mais vous ne disposez de rien d'autre que les deux cruches.

1. Formalisez le problème en termes d'état du système et d'instructions, puis donnez une suite d'instructions permettant d'atteindre une solution.

**Indication :** Comme dans l'exercice précédent, ne cherchez pas à résoudre le problème de tête.

2. Peut-on trouver une suite d'instructions de façon que l'une des deux cruches contienne 1 litre? 2 litres? 3 litres? 4 litres? 5 litres? 6 litres? 7 litres?

**Exercice**

Écrivez un algorithme qui, étant donné un nombre entier  $N$ , décide si  $N$  est ou non le carré d'un autre entier.

Par exemple, si l'entrée est 16, la réponse est oui. Si l'entrée est 42, la réponse est non.

Votre algorithme doit être assez simple pour que vous puissiez l'appliquer vous-même uniquement à l'aide d'un papier et d'un crayon pour poser les opérations (pas de calculatrice).

### Exercice : Nombre mystère

On a fabriqué un robot «intelligent» capable de

- poser des questions dont la réponse est oui ou non;
- effectuer des opérations de base sur les nombres (addition, soustraction, multiplication, division).

Voici un exemple d'algorithme pour le robot :

```
Reponse = Demande("Êtes -vous né avant le 1er janvier 2000 ?")
```

```
Si Réponse = Oui alors
```

```
  Dire("Vous avez plus de 22 ans")
```

```
Sinon
```

```
  Dire("Vous avez 22 ans ou moins")
```

1. Décrivez un algorithme qui permet au robot de deviner un nombre choisi entre 0 et 100 par l'utilisateur.
2. Décrivez un algorithme qui permet au robot de donner l'âge exact de l'utilisateur.
3. Quelles difficultés avez-vous rencontrées pour écrire les algorithmes?

### Exercice : ♣ Un peu de logique et de booléens

1. Construisez les tables de vérité correspondant aux opérations booléennes suivantes :
  - $op1(a,b)$ :  $(a \text{ ET } (NON \ b)) \text{ OU } ((NON \ a) \text{ ET } b)$
  - $op2(a,b)$ :  $(a \text{ OU } b) \text{ ET } ((NON \ a) \text{ OU } (NON \ b))$ .
2. Que constatez-vous?
3.  $op1(a,b)$  est en fait une opération très courante en informatique; elle a même un nom. Savez-vous comment on l'appelle?
4. Écrivez une opération  $op3(a,b)$  dont le résultat est vrai si et seulement si le résultat de  $op1(a,b)$  est faux. Utiliser au plus 5 mots OU, NON et ET dans l'écriture de  $op3(a,b)$ .
5. Expliquez pourquoi cette opération permet de déterminer si deux mots booléens sont identiques.

### Exercice : ♣ Notion de test de programme

Une collègue s'est occupée de coder un programme `pointDeChute` qui calcule l'abscisse à laquelle retombe un projectile lancé en  $x = 0$  avec une vitesse  $v$  suivant un angle  $\alpha$  (en degrés). Vous voulez tester si les résultats de ce programme sont cohérents.

1. Spécifiez le programme sans écrire l'algorithme : que va-t-il prendre en entrée? renvoyer en sortie?
2. Donnez une série de tests dont vous connaissez le résultat sans calculs: situations extrêmes (par ex : tirer verticalement), symétries, ...