

## *Feuille de travaux dirigés n°2*

### Variables, types et instructions

Trois séances

#### Exercice 2.1 (Type de variables)

Pour chacune des informations ci-dessous, indiquer le nombre et le type des variables nécessaires et proposer des noms pour ces variables.

Rappel : le nom de variable doit indiquer ce que la variable contient. Si le nom comporte plusieurs mots, ajouter un tiret bas entre chaque mot (exemple : `taille_moyenne`).

- |                                      |                                   |   |
|--------------------------------------|-----------------------------------|---|
| 1. nombre d'étudiants dans le groupe | 4. une position GPS               | 7. une distance                         |
| 2. un prénom et un nom               | 5. une remarque                   | 8. la position d'un pixel sur l'écran   |
| 3. une température                   | 6. la première lettre d'un prénom | 9. un déplacement de souris sur l'écran |

#### Exercice 2.2 (Simulation : échange de valeurs)

1. Soit la suite d'instructions ci-dessous. Réaliser un historique d'exécution présentant sur chaque ligne l'état des variables à l'issue de l'exécution de chaque instruction. L'utilisateur rentre successivement 1 et 2.

```
variables
  x : Entier
  y : Entier
debut
1  ecrire "donner x"
2  lire x
3  ecrire "donner y"
4  lire y
5  x ← y
6  y ← x
7  ecrire "valeur de x " , x
8  ecrire "valeur de y " , y
fin
```

2. Qu'observez vous ? que faut-il faire pour que l'algorithme échange bien les deux valeurs saisies ?
3. Comment effectuer une permutation circulaire des valeurs de trois variables saisies ?
4. Et pour une permutation circulaire de  $n$  variables ?

#### Exercice 2.3 (Conditionnelle simple et variable booléenne)

Rappel : Il est possible d'utiliser une variable booléenne ou une expression logique à base de variables booléennes dans une expression conditionnelle

1. Dans un premier temps, écrire un algorithme qui initialise (sans demander à l'utilisateur) les variables booléennes `il_vente` et `il_pleut` puis affiche "imperméable" s'il vente et il pleut, "parapluie" si il pleut sans vent et enfin "ni parapluie, ni imperméable" si il ne pleut pas.

2. Modifier cet algorithme pour qu'il commence par demander s'il vente puis s'il pleut. Les humains ne parlant pas le booléen, utiliser un type plus adapté pour obtenir leur réponse avant d'intégrer l'information dans vos variables booléennes.

#### Exercice 2.4 (Conditionnelle : Calcul d'âge)

Ecrire un algorithme qui demande l'année de naissance d'une personne puis calcule et affiche son âge au 31 décembre 2021 à minuit. Par convention, une personne née après cette date aura un âge de 0 ans. Soigner les messages à l'utilisateur. Proposer un second algorithme équivalent.

#### Exercice 2.5 (Conditionnelles multiples : polynôme du second degré)

1. Ecrire un algorithme qui calcule les racines réelles d'un polynôme du second degré après la saisie de ses coefficients ( $a \neq 0$  et  $b \neq 0$  et  $c \neq 0$  pour simplifier). Lors de l'étape d'analyse, utiliser un arbre de décision. L'algorithme affiche les résultats sous forme du texte "pas de solution réelle" ou de la solution double ou des deux solutions réelles.
2. Pour aller plus loin, compléter l'algorithme pour prendre en compte les différents cas de nullité des coefficients et les solutions imaginaires.

#### Exercice 2.6 (Expression booléenne : année bissextile)

Une année est bissextile si (et seulement si) elle est soit divisible par 4 mais pas par 100, soit divisible par 400.

1. Dessiner un arbre de décision utilisant des conditions simples du type "*année divisible par n ?*" permettant de décider si l'année en question est bissextile ou non. L'ordre d'évaluation des prédicats n'a pas d'impact sur le résultat, mais mérite réflexion. Pourquoi ? Proposer un jeu de test.
2. Écrire l'algorithme qui affiche "bissextile" ou "non bissextile" selon l'année saisie par l'utilisateur.
3. Transformer l'arbre de décision en expression booléenne.
4. ré-écrire l'algorithme qui saisit une année et affiche si l'année est bissextile ou non.

#### Exercice 2.7 (Répétitives : course de fond)

Un coureur de fond effectue des tours de stade. Son entraîneur chronomètre ses passages en saisissant le temps de chaque tour en secondes.

1. Effectuer une analyse pour ensuite écrire un algorithme qui demande la saisie du nombre de tours prévu, puis permet la saisie des temps de passage. Après la saisie, le temps total est affiché, ainsi que le temps de passage moyen. Ecrire une version de l'algorithme utilisant une boucle pour et une autre utilisant une boucle tantque. Comparer.
2. Modifier cet algorithme afin de le rendre plus flexible. Le nombre de tours prévu n'est plus demandé. À la place l'entraîneur saisit une valeur drapeau (-1) pour signaler que le dernier temps de passage a été saisi. La valeur -1 n'est pas saisie dès la première invite (le coureur fait au moins un tour).
3. Une dernière amélioration affiche à la fin le temps de passage minimum et le temps maximum.

#### Exercice 2.8 (Répétitives : matrices identités)

Un algorithme doit afficher des matrices identités de taille variable.

Dimension 2 :  $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Dimension 3 :  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Effectuer une analyse et en déduire un algorithme qui, après avoir demandé à l'utilisateur de rentrer la dimension, affiche la matrice correspondante.

#### Exercice 2.9 (Compression RLE)

La compression RLE (*Run-length encoding*) est utilisée par exemple dans le format d'image BMP (un des premiers formats d'images numériques).

Cette compression tire parti de la cohérence dans les données. Par exemple, en RLE, la chaîne 'AAAAABBBBBRRRRRAAA' est codée par la chaîne '5A2B4R3A'. Analyser et écrire un algorithme qui demande à l'utilisateur de saisir une chaîne de caractères, calcule puis affiche la chaîne compressée correspondante.