

PROGRAMMATION 1

TD2

Notions à acquérir**Instructions alternatives :** Complètes**Si Condition****alors** Instructions 1**sinon** Instructions 2**fin si**

Incomplètes

Si Condition**alors** Instructions 1**fin si**Hiérarchie des **priorités** dans une expression booléenne**Exercice 1**

Les expressions suivantes sont-elles bien de type booléen ? (C'est à dire, peut-on utiliser ces expressions dans la condition d'un **Si** ?)

- a) $(x < y)$ et $(z == 4)$ (avec x, y, et z entiers)
b) $(x < y)$ ou $(z$ et $(t == 4))$ (avec x, y, z, et t entiers)

Dans les expressions qui suivent, indiquez les types des variables utilisées afin que ces expressions soient booléennes

- c) x ou (y et $(z < 5)$)
d) x ou (y et $(x < 5)$)

Exercice 2

Comment sont interprétées les instructions suivantes ?

(Réindentez le programme pour y voir plus clair, et expliquer dans quels cas on exécute Instructions_1 ou Instructions_2):

Si Condition_1 **alors si** Condition_2 **alors** Instructions_1 **sinon** Instructions_2 **finsi**

Si Condition_1 **alors si** Condition_2 **alors** Instructions_1 **sinon** Instructions_2 **finsi finsi**

Si Condition_1 **alors si** Condition_2 **alors** Instructions_1 **finsi sinon** Instructions_2 **finsi**

Si Condition_1 **alors si** Condition_2 **alors** Instructions_1 **finsi finsi sinon** Instructions_2

Exercice 3

Les deux (morceaux) de programme suivants sont ils équivalents ?

```
N : entier
Saisir(N)
Si N mod 2 == 1 alors
    N ← 3*N+1
sinon
    N ← N div 2
finsi
```

```
N : entier
Saisir(N)
Si N mod 2 == 1 alors
    N ← 3*N+1
finsi
Si N mod 2 == 0 alors
    N ← N div 2
finsi
```

Exercice 4

Etant donnés quatre nombres entiers, écrire un algorithme qui permet d'afficher :

- 4 si les quatre nombres sont égaux
- 3 si trois des nombres sont égaux et le quatrième distinct
- 2 si deux nombres sont égaux, les deux autres étant distincts des deux premiers et entre eux
- 22 si deux nombres sont égaux, les deux autres étant distincts des deux premiers et égaux entre eux
- 1 si tous les nombres sont distincts

Exercice 5

Etant donnée une suite de nombres entiers positifs, saisie à partir du clavier, et qui se termine par le nombre 0, écrire un algorithme qui calcule le nombre de valeurs paires contenues dans cette suite.

Exercice 6

- a) Ecrire un algorithme qui demande 20 nombres entiers naturels à l'utilisateur, et qui affiche ensuite à l'écran la valeur du plus grand parmi les nombres saisis.
- b) Même question que a), mais on ne connaît pas à l'avance le nombre de valeurs qui seront saisies au clavier.
- c) Même question que a), mais les nombres saisis sont des entiers relatifs (valeurs négatives possibles)

Exercice 7

a)

Ecrire un algorithme qui recherche les racines réelles d'un polynôme de la forme

$$a * x^2 + b * x + c = 0$$

On suppose que a, b, c sont des entiers donnés par l'utilisateur.

b)

Ecrire un algorithme qui recherche les racines entières entre -100 et 100 d'un polynôme de la forme

$$a * x^3 + b * x^2 + c * x + d = 0$$

On suppose que a, b, c, d sont des entiers donnés par l'utilisateur.

S'il existe des racines réelles entre -100 et 100, peut-on donner leur partie entière ?

Exercice 8

Considérons une phrase écrite avec seulement des lettres minuscules et qui est sans ponctuation, mis à part un point final. Cette phrase est constituée de mots séparés les uns des autres par un espace et un seul. La phrase commence obligatoirement par la première lettre du premier mot et se termine par le point final qui est accolé à la dernière lettre du dernier mot.

- a- Ecrire un algorithme qui calcule le nombre de lettres 'm' présentes dans la phrase.
- b- Ecrire un algorithme qui calcule le nombre de mots constituant cette phrase.
- c- Ecrire un algorithme qui calcule le nombre de syllabes 'le' présentes dans la phrase.
- d- Ecrire un algorithme qui calcule la longueur du dernier mot de la phrase.
- e- Ecrire un algorithme qui calcule quelle est la longueur du plus long mot de la phrase.

Exercice 9

Ecrire un algorithme qui additionne les premiers nombres entiers jusqu'à obtenir une somme la plus proche possible d'un nombre Nb saisi en début d'algorithme.

En cas d'écarts relatifs égaux, nous prendrons la somme trouvée par valeur inférieure.

Exercice 10

a) Une année est bissextile si elle est divisible par 4 mais pas par 100, à moins qu'elle ne soit divisible par 400.

Ecrire un algorithme qui saisit une année et affiche "année bissextile" si elle est bissextile, et "année non bissextile" sinon.

b) On dit qu'une date écrite sous la forme de trois entiers (j, m, a), j représentant le jour, m le mois et a l'année, est valide si et seulement si $a > 0$, $1 \leq m \leq 12$ et $1 \leq j \leq \text{nombre de jours du mois}$.

Ecrire un algorithme qui saisit une date et affiche "date valide" si elle est valide, et "date invalide" sinon. Si la date est invalide, l'algorithme affiche aussi l'une des raisons pour lesquelles elle est invalide.

c) Idem b), sauf que si la date est invalide, l'algorithme affiche "date invalide" puis non pas une, mais toutes les raisons pour lesquelles elle est invalide.

Exercice 11

Le but de cet exercice est d'écrire un algorithme qui, pour une date comprise entre le 1^{er} janvier 1901 et le 28 février 2100 écrite sous la forme de trois entiers (j, m, a), j représentant le jour, m le mois et a l'année, calcule le nombre de jours écoulés entre le 1^{er} janvier 1901 et cette date.

Noter que dans cette période, les années bissextiles sont exactement les multiples de 4.

1. Ecrire un algorithme qui, à partir d'une date $d = (j, m, a)$ de cette période, calcule le nombre de jours écoulés entre le 1^{er} janvier de l'année a et la date d (ces 2 jours compris).

Par exemple, si $d = (5, 3, 2016)$ le nombre de jours écoulés est $31 + 29 + 5 = 65$

et si $d = (5, 3, 2018)$ le nombre de jours écoulés est $31 + 28 + 5 = 64$

2. Même question en écrivant en langage algorithmique l'algorithme suivant:

- calculer $Nbe = \text{partie_entiere}(30.42 * (m - 1)) + j$,
- si $m = 2$ (février), augmenter ce nombre de jours Nbe d'une unité.
- si $m > 2$ et $m < 8$ (mars, avril, mai, juin et juillet) diminuer ce nombre Nbe d'une unité.
- si l'année est bissextile et $m > 2$, augmenter ce nombre Nbe d'une unité.

3. **(TP)** Vérifier que les 2 algorithmes précédents calculent le même nombre de jours pour différentes dates saisies au clavier.

4. **(TD et TP)** Pour vérifier de façon plus sûre la correction de chacun des 2 algorithmes, programmer le calcul du nombre de jours pour chacune des dates allant du 1^{er} janvier au 31 décembre d'une année donnée, et vérifier par programme que la suite des résultats obtenus est bien 1, 2, 3 etc. Tester chacun des 2 algorithmes sur une année bissextile et sur une année non bissextile.

5. Ecrire un algorithme qui demande à l'utilisateur de saisir une date d dans cette période et affiche le nombre de jours écoulés entre le 1^{er} janvier 1901 et la date d (ces deux jours compris).

6. Ecrire un algorithme qui demande à l'utilisateur de saisir sa date de naissance et la date d'aujourd'hui et affiche le nombre de jours écoulés depuis sa naissance (jour de naissance et jour d'aujourd'hui compris).