

Traitements conditionnés

Dans beaucoup de situations, des actions ne sont à effectuer que si certaines conditions sont remplies :

- Une équation du second degré a 0, 1 ou 2 solutions réelles suivant que le discriminant soit négatif, nul ou positif.
- Si une quantité d'articles en stock tombe en dessous d'un certain seuil, il faut émettre une commande.
- S'il fait beau alors on va à la plage, sinon on fait de l'algorithmique ensemble.
- S'il ne fait pas beau alors on fait de l'algorithmique ensemble.

Chacun de ces choix est conditionné par une expression booléenne qui, lorsqu'elle est vraie, entraîne l'exécution du traitement associé.

La syntaxe algorithmique pour programmer des traitements conditionnés est des plus simples :

Si Condition

Alors

T1

[**Sinon**

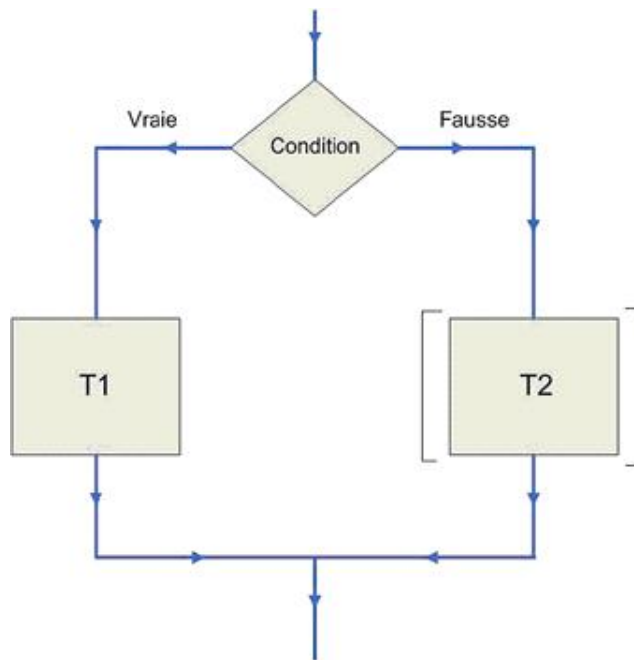
T2]

Finsi

Juste quelques remarques sur cette structure très intuitive :

- Condition est une expression à résultat booléen (Vrai ou Faux).
- T1 et T2 sont des traitements.
- **Finsi** signifie la fin du si.
- Une forme simplifiée, sans l'alternative, existe (sans **Sinon** T2).
- Le test sur la condition se fait sur une hypothèse de condition vraie.
- Le tracé d'un trait vertical (à gauche) entre **Si** et **Finsi** sera un moyen mnémotechnique pour ne pas oublier le **Finsi**.
- Pensez à indenter les traitements T1 et T2.

L'ordinogramme correspondant à cette structure est le suivant :



Dans l'exemple présenté ci-après, une variable de nom NBLU est affectée d'une valeur saisie au clavier et l'algorithme précise ensuite s'il s'agit d'une valeur paire ou impaire :

Début

Co Déclarations **Fco**

Ent NBLU

Co Saisie clavier **Fco**

Ecrire("Nombre : ")

NBLU <- **Lire**

Co Détermination de la parité **Fco**

Si NBLU **Mod** 2 = 0

Alors

Ecrire(NBLU, " est pair")

Sinon

Ecrire(NBLU, " est impair")

Finsi

Fin

1. Exercice n°6 : Polynôme du second degré

Sujet

Calculer les racines d'un polynôme du second degré Ax^2+Bx+C (avec $A \neq 0$ dans l'absolu mais ce test ne sera pas effectué ici). Les valeurs A, B et C seront saisies au clavier.

Corrigé

Début

Co Déclarations **Fco**

Réel A, B, C, DELTA

Co Saisie des paramètres **Fco**

Ecrire("A : ")

A <- **Lire**

Ecrire("B : ")

B <- **Lire**

Ecrire("C : ")

C <- **Lire**

Co Calcul du discriminant **Fco**

DELTA <- (B * B) - (4 * A * C)

Co Détermination du nombre de racines **Fco**

Si DELTA < 0

Alors

Ecrire("Pas de solutions")

Sinon

Si DELTA = 0

Alors

Ecrire("Solution unique = ", -B / (2 * A))

Sinon

Ecrire("Deux racines : ", (-B+DELTA**0.5)/(2*A), " et ", (-B-DELTA**0.5)/(2*A))

Finsi

Finsi

Fin

L'intérêt de cet exercice, outre celui de vous faire réviser un peu les mathématiques, est de montrer qu'il est tout à fait possible d'imbriquer des structures conditionnelles. La mise en œuvre n'est pas difficile mais demande toutefois un minimum de rigueur, notamment ne pas oublier de fermer vos structures par un **Finsi** et aussi d'indenter (ajouter un niveau de décalage) les traitements inclus après les **Alors** et les **Sinon**.

2. Exercice n°7 : Libellé du mois en clair

Sujet

Imprimer en lettres le mois correspondant à un numéro donné au clavier (compris entre 1 et 12). Le contrôle de la saisie n'est pas à prévoir.

Corrigé

Début

Co Déclarations **Fco**

Ent MOIS

Co Saisie du numéro du mois **Fco**

Ecrire("Numéro du mois (1 à 12) : ")

MOIS <- **Lire**

Co Affichage du résultat **Fco**

Si MOIS=1

Alors

Ecrire("Janvier")

```

Sinon

    Si MOIS=2

        Alors

            Ecrire("Février")

        Sinon

            ...

    Finsi

Fin

```

Évidemment la solution proposée est contraignante au niveau de son élaboration mais du point de vue technique elle est optimale. Par exemple la saisie du chiffre 2 donnera le bon affichage ("Février") au bout de deux tests.

Il aurait aussi été possible de "superposer" 12 tests successifs comme ci-après. Cette solution est peu satisfaisante car quelle que soit la valeur saisie au clavier, les 12 tests sont exécutés.

```

Début

    Co Déclarations Fco

    Ent MOIS

    Co Saisie du numéro du mois Fco

    Ecrire("Numéro du mois (1 à 12) : ")

    MOIS <- Lire

    Co Affichage du résultat Fco

    Si MOIS=1

        Alors

            Ecrire("Janvier")

        Finsi

    Si MOIS=2

        Alors

            Ecrire("Février")

        Finsi

    ...

    Si MOIS=12

        Alors

            Ecrire("Décembre")

        Finsi

Fin

```

La structure de contrôle **Suivant ... Finsuivant** est une solution élégante pour gérer ce type de problématique. La syntaxe est la suivante :

```

Suivant VARIABLE/EXPRESSION

    Valeur_1 : Action_1

    ...


    Valeur_n : Action_n

    [Autrecas : Action par défaut]

Finsuivant

```

Dans le cas où la variable VARIABLE (ou l'expression EXPRESSION) est égale à Valeur_1 le traitement Action_1 est déclenché (idem pour Valeur_2 avec dans ce cas l'exécution de Action_2 et ainsi de suite).

 Dans certains langages de programmation, l'implémentation de la structure **Suivant ... Finsuivant** est un peu particulière. Après la détection de la condition vraie, toutes actions associées aux conditions suivantes sont exécutées que ces conditions soient vraies ou fausses. Dans ces langages, il faut prévoir d'ajouter un mot clé particulier provoquant le renvoi direct au **Finsuivant**.

La structure **Suivant ... Finsuivant** (ou son équivalent) n'est pas implémentée dans tous les langages de programmation.

3. Exercice n°8 : Libellé du mois en clair (Suivant ... Finsuivant)

Début

Co Déclarations **Fco**

Ent MOIS

Co Saisie du numéro du mois **Fco**

Ecrire("Numéro du mois (1 à 12) : ")

MOIS <- **Lire**

Co Affichage du résultat **Fco**

Suivant MOIS

1 : **Ecrire**("Janvier")

2 : **Ecrire**("Février")

...

12 : **Ecrire**("Décembre")

Autrecas : **Ecrire**("Erreur de saisie du n° de mois")

Finsuivant

Fin

Dans cet exercice un message d'erreur est affiché dans le cadre du **Autrecas**. Par contre, il faudra que l'utilisateur de l'algorithme (ou du programme) le relance car aucun dispositif n'est pour l'instant prévu pour qu'une nouvelle saisie soit proposée.