

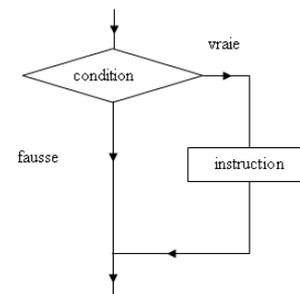
Structures conditionnelles

En algorithmique, on est très souvent amené à effectuer des instructions sous certaines conditions. Les conditions sont des expressions logiques qui peuvent être vraies ou fausses (par exemple la comparaison de deux valeurs). Pour effectuer un test conditionnel, on utilise une boucle `if`. Cette boucle doit toujours être fermée par un `end`. Il y a trois syntaxes différentes pour une boucle `if`.

1 Version courte : `if...then...end`

Dans ce cas, on fait un seul test. Si la *condition* est vraie, les *instructions* entre le `then` et le `end` de la boucle sont effectuées. Si la *condition* est fausse, les *instructions* entre le `then` et le `end` de la boucle sont ignorées.

```
if condition then
    instructions
end
```



Exercice 1 1. Entrer dans l'éditeur de Scilab les instructions suivantes :

```
a=input('Donner une valeur non nul pour a: ')
b=input('Donner une valeur pour b: ')
c=input('Donner une valeur pour c: ')
x=b^2-4*a*c
if x>0 then
    disp('Le discriminant est strictement positif.')
    x1=(-b-sqrt(x))/(2*a)
    x2=(-b+sqrt(x))/(2*a)
    disp(x1, 'x1=')
    disp(x2, 'x2=')
end
```

2. Enregistrer et exécuter. Tester avec les polynômes suivants :

$$P_1(x) = x^2 - 4x + 4,$$

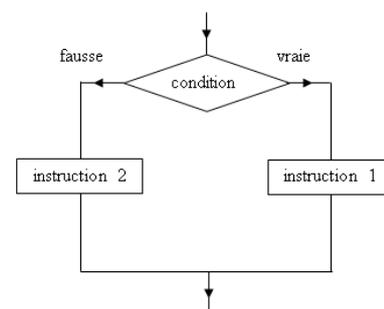
$$P_2(x) = x^2 + 5x + 4,$$

$$P_3(x) = 2x^2 + x + 2.$$

2 Version intermédiaire : `if...then...else...end`

Ici aussi, on fait un seul test. Si la *condition* est vraie, le premier bloc d'instructions (*instructions₁*) entre le `then` et le `else` de la boucle est effectué. Si la *condition* est fausse, le deuxième bloc d'instructions (*instructions₂*) entre le `else` et le `end` de la boucle est effectué.

```
if condition then
    instructions1
else
    instructions2
end
```



Exercice 2 Construire une suite d'instructions qui étant donnés deux réels a et b , donne le maximum de a et de b . Notons que Scilab dispose de fonctions `max` et `min` qu'on n'utilisera pas ici.

Exercice 3 Tracer sur l'intervalle $[-2, 5]$ la fonction définie par :

$$f(x) = \begin{cases} \sqrt{x} & \text{si } x \geq 0, \\ x^2 & \text{si } x < 0. \end{cases}$$

3 Version longue : `if...then...elseif...else...end`

Dans ce cas, on fait plusieurs tests. Si la $condition_1$ est vraie, le premier bloc d'instructions ($instructions_1$) est effectué. Si la $condition_1$ est fausse, on fait un deuxième test. Si la $condition_2$ est vraie, le deuxième bloc d'instructions ($instructions_2$) est effectué. Ce procédé est répété jusqu'à la $condition_{k-1}$. Si la $condition_{k-1}$ est vraie, le bloc d'instructions ($instructions_{k-1}$) est effectué. Si la $condition_{k-1}$ est fausse, le dernier bloc d'instructions ($instructions_k$) est effectué.

```

if condition1 then
    instructions1
elseif condition2 then
    instructions2
...
elseif conditionk-1 then
    instructionsk-1
else
    instructionsk
end

```

Exercice 4 1. Modifier les instructions données dans l'exercice 1 pour que, étant donnés les trois coefficients a, b, c d'un polynôme $P(X) = aX^2 + bX + c$, on ait en sortie les deux racines distinctes de P (si le discriminant est strictement positif) ou l'unique racine double de P (si le discriminant est nul) ou le message " P n'admet pas de racine" (si le discriminant est strictement négatif).

Remarque. Pour tester une égalité sur Scilab, on utilise la commande `==`.

2. Enregistrer et exécuter. Tester avec les polynômes donnés à l'exercice 1.

Exercice 5 Construire une suite d'instructions avec en entrée la moyenne générale m d'un étudiant de classe préparatoire et en sortie le message :

- "Passage en classe supérieure avec les félicitations du conseil de classe" si $m \geq 14$,
- "Passage en classe supérieure avec les encouragements du conseil de classe" si $14 > m \geq 12$,
- "Passage en classe supérieure" si $12 > m \geq 10$,
- "Redoublement" si $10 > m$.

Exercice 6 Écrire une suite d'instructions simulant la saisie du code confidentiel d'une carte bancaire : le code confidentiel est une donnée du programme; l'utilisateur a trois essais pour taper le bon code.