

TP3 : Les branchements Conditionnels

1 L'instruction : if

L'instruction '*if*' est employée lorsqu'on veut exécuter un bloc d'instructions que lorsque certaines conditions sont vraies. Si ces conditions ne sont pas vérifiées alors ce bloc d'instructions ne sera jamais exécuté.

Exemple : la division par zéro.

Algorithmique	Langage C
Si <expression> alors <Instructions à exécuter si l'expression est vraie> ; Fsi ;	if (<expression>){ <Instructions à exécuter si l'expression est vraie> ; }

2 L'instruction : if ... else

Cette instruction est utilisée lorsqu'on a deux blocs à exécuter et sous une certaine condition, on exécute soit l'un soit l'autre. Les deux blocs ne seront jamais exécutés en même temps.

Exemple : Un nombre est soit pair ou impair.

Algorithme	Langage C
Si <expression> alors <Instructions à exécuter si l'expression est vraie> ; Sinon <Instructions à exécuter si l'expression est fausse> ; Fsi ;	if (<expression>){ <Instructions à exécuter si l'expression est vraie> ; } else{ <Instructions à exécuter si l'expression est fausse> ; }

Parfois nous aurons à évaluer des expressions et selon leurs valeurs de vérité, nous exécuterons de différentes instructions. Pour ce faire, on procède comme suit :

Algorithme	Langage C
Si <expression_1> alors <Instructions à exécuter si l'expression est vraie> ; Sinon Si <expression_2> alors	if (<expression_1>){ <Instructions à exécuter si l'expression est vraie> ; } else if (<expression_2>){

<Instructions à exécuter si l'expression_1 est fausse et l'expression_2 est vraie > ; Sinon <Instructions à exécuter si toutes les expressions sont fausses> ; Fsi ; Fsi ;	<Instructions à exécuter si l'expression_1 est fausse et l'expression_2 est vraie > ; } else{ <Instructions à exécuter si toutes les expressions sont fausses> ; }
--	--

Exemple : soient deux nombre a et b , nous avons soit $a < b$, $b > a$ ou $a = b$.

Exemple : Le programme suivant donne la catégorie d'âge de l'utilisateur sachant son âge.

```

main.c ✕
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int age;
7      printf("***** Catégories d'âge: *****\n");
8      printf("Quel est votre âge?\n");
9      scanf("%d", &age);
10     printf("Vous avez %d |ans, vous êtes ", age);
11
12     if(age<9)
13     {
14         printf("jeune minibad");
15     }
16     else if(age<11)
17     {
18         // ici l'âge est 9<= age < 11
19         printf("jeune poussin");
20     }
21     else if(age<13) printf("jeune benjamin");
22     else if(age<15) printf("jeune minime");
23     else if(age<17) printf("jeune cadet");
24     else if(age<19) printf("jeune junior");
25     else if(age<40) printf("adulte senior");
26     else if(age>=40) printf("adulte vétéran");
27
28     printf(".\n");
29     //system("PAUSE");
30     return 0;
31 }

```

Son exécution donne :

```

example_1_ifelse
***** Catégories d'âge: *****
Quel est votre age?
19
Vous avez 19 ans, vous êtes adulte senior.

Process returned 0 (0x0)   execution time : 4,319 s
Press ENTER to continue.

```

3 L'instruction : switch ... case

Lorsqu'on a des blocs à exécuter selon une certaine valeur d'une expression ou une variable, l'instruction *switch ... case* est employée.

Algorithme	Langage C
Cas <expression> vaut : constante_1 : <instructions> ; cas constante_n : <instructions> ; sinon <instructions> ; Fcas ;	<pre> switch (<expression>){ case constante_1 : <instructions> ; break ; case constante_n : <instructions> ; break ; default : <instructions> ; } </pre>

Exercices

1. Écrire un programme qui calcule toutes les solutions possibles d'une équation quadratique si elles existent (dans R).

Une équation quadratique est de la forme :

$$2. \quad ax^2 + bx + c = 0, \text{ avec } a \neq 0.$$

Pour trouver toutes les solutions possibles dans R, il faut d'abord calculer : $\Delta = b^2 - 4ac$,

Si $\Delta < 0$: alors l'équation quadratique n'a pas de solution dans R,

sinon, si $\Delta = 0$ alors l'équation quadratique a une solution double ($x = \frac{-b}{2a}$),

sinon, elle admet deux solutions x_1 et x_2 qui sont :

$$3. \quad x_1 = \frac{-b + \sqrt{\Delta}}{2a} \text{ et } x_2 = \frac{-b - \sqrt{\Delta}}{2a} .$$

4. Programmer une calculatrice qui effectue initialement les opérations suivantes : l'addition, la soustraction, la multiplication, la division et le modulo.
Vous pouvez rajouter d'autres opérations comme la puissance et la racine carrée.