

TD 2 — Structures conditionnelles en C

Dans ce TD, nous allons étudier la syntaxe du langage C pour les structures conditionnelles étudiées dans la ressource R1.13 (Méthodologie, réussite universitaire).

Travail à rendre

Ce TD est constitué de trois parties, chaque partie contient plusieurs exercices. Vous devrez rendre une archive R101_TD2_<nomDeFamille>.zip, contenant trois projets Code::Blocks : un pour chaque partie.

Chaque projet contiendra les réponses à tous les exercices de la partie correspondante. Vous veillerez à ce que les affichages dans la console permettent de facilement se repérer à travers les exercices.

1 Structures conditionnelles élémentaires

En langage C, la structure conditionne SI ALORS... SINON suit la syntaxe suivante :

```
1  if(condition) { /* condition */
2      /* Instructions si la condition est vérifiée */
3  }
4  else { // Facultatif
5      /* Instructions si la condition n'est pas vérifiée */
6  }
```

Attention, il n'y a pas de point-virgule après les accolades.

Exercice 1. Écrire un programme qui calcule et affiche le minimum de deux variables.

Exercice 2. Écrire un programme qui demande la saisie au clavier d'une heure, au moyen de trois variables *h* (heures), *min* (minutes) et *s* (secondes). Afficher l'heure qu'il sera une seconde plus tard, en envisageant tous les cas possibles.

Exercice 3. Écrire un programme qui demande la saisie de deux flottants *a* et *b* (c'est-à-dire ayant un signe positif ou négatif) et affiche le *signe* du produit *a*b* sans le calculer.

Exercice 4. Écrire un programme qui affiche à l'écran si un nombre entier saisi au clavier est pair ou impair.

L'opérateur modulo

L'opérateur modulo (noté % en C) pourra être utile. Si *a* et *b* sont deux variables entières avec *b* non nulle, alors *a%b* est le reste dans la division euclidienne de *a* par *b*.

2 Comparaison à des valeurs entières et opérateur ternaire

2.1 La structure switch

La structure `switch` permet de tester si une variable est *égale* à des *constantes entières*, lorsqu'il y a un nombre fini de valeurs possibles. C'est un cas typique d'usage pour créer des menus.

```
1  int moyenDeTransport = 0;
2  printf("Comment venez-vous à l'IUT?\n");
3  printf("\t 1. Transports en commun.\n"); // \t : tabulation
4  printf("\t 2. Marche à pied.\n");
5  printf("\t 3. Vélo.\n");
6  printf("\t 4. Voiture.\n");
7  printf("\t 5. Autre\n");
8  printf("Réponse : ");
9  scanf("%d", &moyenDeTransport); // Valeur saisie au clavier
10
11 switch(moyenDeTransport) {
12     case 1: // Transports en commun
13         printf("Félicitations, c'est bon pour la planète !");
14         break; // Pour sortir du switch.
15     case 2: // Marche à pied
16     case 3: // Vélo
17         // On gère les cas 2 et 3 en même temps
18         printf("Bravo, c'est bon pour la santé et la planète.");
19         printf("Le vélo est le plus rapide en ville");
20         break; // Pour sortir du switch.
21     case 4:
22         printf("N'avez-vous pas d'autre alternative ?\n");
23         break; // Pour sortir du switch.
24     case 5:
25         printf("Vous venez en tapis volant ?!");
26         break;
27     default: // N'importe quel autre valeur : erreur de saisie.
28         printf("Il doit y avoir une erreur...");
29 }
```

Exercice 5. Écrire un programme qui demande à l'utilisateur ou utilisatrice de saisir un chiffre et qui l'écrit ensuite en toutes lettres à l'écran. Par exemple, si l'on saisit 9, le programme affichera « neuf ». Afficher un message d'erreur si une valeur n'appartenant pas à $\llbracket 0, 9 \rrbracket$ est saisie.

Exercice 6. Écrire un programme qui demande le numéro du mois et affiche le nombre de jours qu'il contient (on considère que le mois de février compte 28 jours).

2.2 L'opérateur ternaire ?

Il est fréquent qu'une condition consiste uniquement à affecter une variable selon la valeur logique d'une expression booléenne. La structure du code serait alors la suivante :

```

1  if (a>b) // Expression logique
2      max = a; // Affectation de la variable si l'expression vaut true
3  else
4      max = b; // Affectation de la variable si l'expression vaut false

```

Le langage C propose alors une syntaxe plus courte et plus lisible :

```

1  max = (a>b) ? a : b;

```

La syntaxe est donc `variableAffecter = (test) ? valeurSiVrai : valeurSiFaux;`

Cet opérateur noté avec un point d'interrogation est un *opérateur ternaire* car il prend trois opérandes : le test et les deux valeurs à affecter.

Exercice 7. Compléter le programme qui demande à l'utilisateur ou utilisatrice de saisir le nombre d'enfants. Ensuite, le programme affiche par exemple : « Vous avez 7 enfants. » Le mot « enfant » doit être au pluriel si le nombre saisi est strictement supérieur à 1.

```

1  int nbEnfants = 0;
2
3  /* Saisie le nombre d'enfants */
4  printf ("Combien d'enfants ? ");
5  scanf ("%d", &nbEnfants);
6
7  printf ("Vous avez %d enfant%c.\n", nbEnfants, /* COMPLÉTEZ ICI */ );

```

Astuce : le caractère `'\0'`

Le caractère spécial `'\0'` est le caractère... vide, qui n'affiche rien à l'écran. Sa valeur ASCII est 0. Attention à ne pas le confondre avec le caractère `'0'` (de type `char`), qui est le *texte* du chiffre zéro. À distinguer également du *nombre* 0 (de type `int`).

3 Quelques exercices d'application

Exercice 8. Une agence loue des vélos 250 € la première semaine, auxquels il faut rajouter 25 € pour chaque jour supplémentaire entre le 8^e et le 18^e jour, et 17 € pour chaque jour suivant.

- 1) Écrire un programme permettant d'établir le prix de la location en fonction du nombre de jours loués.
- 2) On souhaite décider s'il serait préférable d'acheter un vélo. Ajouter des lignes de code permettant de saisir la valeur du vélo que l'on souhaite acheter, puis qui calcule le nombre de jours de location à partir duquel l'achat devient plus économique.

Exercice 9. Ecrire un programme qui lit un caractère et détermine s'il fait partie des alphabets ou non. Et s'il l'est, dire en plus s'il est une minuscule ou une majuscule.

Indication — On commence par la lecture du caractère vers une variable de type `char`, à l'aide la fonction `scanf`. En utilisant le format dédié à ce type qui est `%c`. En langage C, les caractères

sont traités comme des entiers. On peut donc comparer deux caractères sans aucun problème. Par exemple, on a `'a' < 'b'`.

À retenir

- La principale structure conditionnelle est la structure **if else**.
- Il est possible de constituer des structures conditionnelles plus complexes :

```
1  if(test1) {  
2      // Instructions  
3  }  
4  else if(test2) {  
5      // Instructions  
6  }  
7  else if(test3) {  
8      // Instructions  
9  }  
10 else {  
11     // Instructions  
12 }
```

- Lorsqu'il n'y a qu'un nombre fini de valeurs entières à tester, on utilise la structure **switch**.
- Pour les tests booléens permettant d'affecter une valeur à une variable, l'opérateur ternaire `?` peut être utilisé.