



– Algorithmique – Travaux Pratiques – Série N°2

I – Nouveaux Mots Clés : En plus des mots clés traités dans la série de TP N°1 (*char, short, int, long, float, double, const, signed, unsigned*), s'ajoutent les nouveaux mots clés suivants : *if, else, switch, case, break, default*.

II – Les structures conditionnelles en algorithmiques et en langage C

Les structures conditionnelles permettent de traiter les différents cas d'un problème.

La structure de base est la conditionnelle (alternative) complète qui permet de traiter deux cas :

Si (condition) Alors sequence1 // Un cas	Traduction en langage C	→→→	if (condition) { sequence1 } else { sequence2 }
Sinon sequence 2 // Les autres cas			
FSi			

Rappel : Condition est une expression booléenne ; Séquence est une suite d'instructions (simples ou structurées). En langage C, une séquence commence par '{' (Début) et se termine par '}' (Fin).

✓ Si la sequence2 est vide, on utilise la conditionnelle (l'alternative) réduite :

Si (condition) Alors sequence1	Traduction en langage C	→→→	if (condition) { sequence1 }
FSi			

✓ Si une séquence est formée d'une seule instruction, les accolades { } deviennent facultatives.

Exemple : Si **a** est une variable de type **int**, on peut écrire par exemple :

```
if (a < 0)
    printf("%d est negatif\n", a) ;
else
    printf("%d est positif\n", a) ;
```

L'opérateur ternaire : Si une structure conditionnelle est de la forme :

<pre>if (condition) variable1 = expression1 ; else variable1 = expression2 ;</pre>	On peut l'écrire : variable1 = (condition) ? expression1 : expression2 ;
--	--

Exemple : Si **a, b** et **max** sont des nombres de type **int** et on a la structure conditionnelle suivante :

<pre>if (a > b) max = a ; else max = b ;</pre>	on peut l'écrire tout simplement : max = (a > b) ? a : b ;
---	--

✓ Pour traiter plus de deux cas (c'est-à-dire vérifier plusieurs conditions), il faut utiliser des structures conditionnelles imbriquées. **Exemple** : Pour résoudre une équation du second degré, on peut traiter les trois différents cas du discriminant (delta) comme suit :

Si delta < 0 Alors

 Ecrire("Pas de solution")

Sinon

 Si delta = 0 Alors

 Ecrire("Solution double")

 Sinon

 Ecrire("Deux solutions")

 FSi

FSi

Traduction en
langage C

```
if (delta < 0)
    printf("Pas de solution\n");
else if (delta == 0)
    printf("Solution double\n");
else
    printf("Deux solutions\n");
```

✓ Si le nombre de cas à traiter est important, il faut utiliser une suite de blocs : if, else if, else if, ..., else pour tester toutes les conditions.

✓ Si toutes les conditions sont des conditions d'égalité entre la même expression et une liste de constantes, on peut simplifier l'écriture en utilisant la structure conditionnelle à choix multiples (**Selon**) :

Selon (expression) Vaut

 cste1 : sequence1

 cste2, cste3 : sequence2

 cste4, cste5, cste6 : sequence3

 ⋮

 Autre : sequenceN

FinSelon

switch (expression)

{

case cste1 : sequence1;

break ; // Une seule constante

case cste2 :

case cste3 : sequence2;

break ; // Liste de deux constantes

case cste4 :

case cste5 :

case cste6 : sequence3;

break ; // Liste de plusieurs constantes

 ⋮

 // Les autres cas non traités

Default : sequenceN ;

}

Remarques sur l'instruction switch :

✓ Une constante ne peut apparaître qu'une seule fois.

✓ Si plusieurs constantes partagent la même séquence d'instructions, il faut les écrire dans des clauses "**case**", les unes à la suite des autres, et de n'écrire la séquence qu'en face de la dernière constante.

✓ L'instruction **break** permet de séparer les différentes séquences d'instructions associées aux différentes listes de constantes. En fait, chaque séquence peut être, ou ne pas être, entourée avec des accolades { }.

III – EXERCICES – LES STRUCTURES CONDITIONNELLES

PARTIE 1

1- Écrire un programme C qui détermine si un entier positif **n** est pair ou impair.

Rappel : L'opérateur % permet de calculer le reste de la division entière. **Exemple** : $7 \% 2$ donne 1.

2- Écrire un programme C qui lit deux entiers **a** et **b** et affiche leur minimum (et aussi leur maximum).

3- Écrire un programme C permettant de résoudre l'équation : $\mathbf{aX} + \mathbf{b} = \mathbf{0}$. **a** et **b** sont deux nombres réels. Il faut traiter le cas où **a** est nul ($\mathbf{a} = \mathbf{0}$).

4- Écrire un programme C permettant de résoudre l'équation $\mathbf{aX}^2 + \mathbf{bX} + \mathbf{c} = \mathbf{0}$. **a**, **b** et **c** sont des nombres réels. Pour obtenir la racine d'une variable x de type réel, on doit utiliser la fonction **sqrt** définie dans la bibliothèque **math**. Pour cela, il faut ajouter la ligne **#include <math.h>** au début du programme source.

5- Une année est **bissextile** (comptant 366 jours au lieu de 365) si elle est divisible par 4 mais pas par 100 sauf si elle est multiple de 400.

Par exemple, 2000 et 2008 sont des années bissextiles ; 2006 et 2100 n'en sont pas.

Écrire un programme C permettant de lire une année et affiche ensuite un message indiquant si l'année saisie est bissextile ou pas. Avec votre programme vérifier si les années suivantes sont bissextiles ou non : 2000, 2006, 2008, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020.

6- Une librairie facture 5 DA les dix premières photocopies, 4 DA les vingt suivantes et 3 DA au-delà. Ecrire un programme C qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

PARTIE 2 : SUITE **if, else if, else if, ..., else**

Écrire un programme C qui lit la moyenne générale obtenue par un(e) étudiant(e) et affiche la mention correspondante :

- Si moyenne < 10, mention : **Insuffisant**.
- Si $10 \leq$ moyenne < 12, mention : **Passable**.
- Si $12 \leq$ moyenne < 14, mention : **Assez Bien**.
- Si $14 \leq$ moyenne < 16, mention : **Bien**.
- Si $16 \leq$ moyenne < 18, mention : **Très Bien**.
- Si $18 \leq$ moyenne \leq 20, mention : **Excellent**.
- Si moyenne < 0 ou moyenne > 20, il faut afficher un message d'erreur !

Partie 3 : switch

Une méthode pour trouver le jour (J) de la semaine d'une date (jj/mm/aaaa) est décrite ci-dessous :

Si $mm \geq 3$, $J = \{ [(23mm)/9] + jj + 4 + aaaa + [z/4] - [z/100] + [z/400] - 2 \} \bmod 7$

Si $mm < 3$, $J = \{ [(23mm)/9] + jj + 4 + aaaa + [z/4] - [z/100] + [z/400] \} \bmod 7$

J = Jour de semaine (J = 0 à 6; 0 = Dimanche, 1 = Lundi, ... , 6 = Samedi)

mm = Mois (mm = 1 à 12; 1 = Janvier, 2 = Février, 3 = Mars, ... , 12 = Décembre)

Si $mm < 3$ alors $z = aaaa - 1$

Si $mm \geq 3$ alors $z = aaaa$

/ est l'opérateur de division entière

jj = Jour (jj = 1 à 31)

aaaa = année

mod 7 = Modulo 7= Reste de la division entière par 7

Exemple : Quel jour de la semaine tombait le 01 novembre 1954 ? Réponse : c'est un **Lundi**.

jj = 1, mm = 11, aaaa = 1954 et z = aaaa = 1954 (parce que $mm \geq 3$); d'où :

$J = \{ [(23 \times 11)/9] + 1 + 4 + 1954 + [1954/4] - [1954/100] + [1954/400] - 2 \} \bmod 7 \rightarrow J = 1 \rightarrow$ un **Lundi**.

Q1) Écrire un programme C qui permet de trouver le jour de la semaine d'une date.

Q2) Compléter le programme précédent pour qu'il affiche le nom du jour correspondant.

C'est-à-dire 0 \rightarrow "Dimanche", 1 \rightarrow "Lundi", ... , 6 \rightarrow "Samedi".

OPTIONNEL - Partie 4 : switch

Écrire un programme qui affiche le mois équivalent à un numéro compris entre 1 et 12. C'est-à-dire 1 \rightarrow "Janvier", 2 \rightarrow "Février", ..., 12 \rightarrow "Décembre". Pour n'importe quel autre nombre, il faut afficher un message d'erreur.

IV – Astuce : Pour mettre en forme (indenter) le code source :

1- Sélectionner la partie du code à mettre en forme : soit en utilisant la souris, soit au clavier en appuyant sur shift + les touches de direction. Pour sélectionner le fichier en entier, utiliser le raccourcis **Ctrl+A**.

2- Cliquez droit avec la souris et choisissez **Format use AStyle**. Le code sera indenté automatiquement.

V – Correction De Quelques Erreurs

1- "**error: expected '(' before ...**" *TRADUCTION* "parenthèse ouvrante '(' manquante" : Cette erreur est due, en général, à l'écriture de la condition sans **les parenthèses qui sont obligatoires**.

2- "**error: 'else' without a previous 'if'**" *TRADUCTION* "partie else sans une partie if qui la précède" :

Cette erreur est due, en général, à une partie '**if**' oubliée ou mal-écrite à cause de :

①- La partie '**if**' n'existe pas. ②- Un point-virgule ";" est ajouté après la condition. **Exemple :** if (condition); erreur !

③- Plusieurs instructions sont écrites dans la partie '**if**' sans les accolades { }.

Exemple : if (condition)

instruction1 ;

instruction2 ;

else ...

if (condition)

{ instruction1 ;

instruction2 ; }

else ...

Erreur !

\rightarrow Correction