

# CCI - Principes des langages de programmation

## TD 3 : Branchements conditionnels

François Yvon\*, Thomas Tang†

26 septembre 2007

### Première partie

## Pour bien assimiler ...

Un branchement conditionnel permet de diriger le programme vers un bloc d'instructions particulier suivant le résultat d'un test effectué.

### 1 Branchements simples et branchements imbriqués : if

En métalangage, l'alternative à deux branches s'écrit :

```
SI [condition]  ALORS  [Traitement A]
SINON
                  [Traitement B]
FIN DE SI
```

En langage C, on a alors :

```
if(condition){
    Traitement A
}
else {
    Traitement B
}
```

- "condition" est une expression qui renvoie un entier :
  - différent de zéro si la condition est réalisée,
  - nul sinon
- Traitements A et B peuvent être une instruction seule, ou un bloc d'instructions. Ne pas oublier les points-virgules à la fin de chaque instruction.

La branche SINON (else) peut ne pas exister.

#### 1.1 Exercice 1

*Ecrire un programme qui accepte un entier en entrée et affichez "supérieur à 0" sur la sortie standard si celui-ci est supérieur à 0 et "inférieur ou égal à 0" sinon. Utiliser les fonctions `getchar()` et `scanf()` successivement.*

---

\*yvon@limsi.fr

†tang@cgm.cnrs-gif.fr

## 1.2 Exercice 2

*Ecrire un programme qui demande un caractère en entrée.*

*Si le caractère entré est 'a' ou 'A', demandez lui de rentrer son âge et affichez "Votre âge en secondes est : ". Si le caractère entré est 'p' ou 'P', demandez lui de rentrer son poids et affichez son poids s'il est inférieur à 70, "Il faut manger 5 fruits et légumes par jour" sinon. Attention aux nombre négatifs !*

## 2 Branchements multiples : switch

La construction switch permet d'effectuer différentes actions en fonction des valeurs possibles d'une expression. La syntaxe est :

```
switch ([expression]) {
    case [valeur1] : [liste d'instructions] break ;
    case [valeur2] : [liste d'instructions] break ;
    .
    case [valeur18]:
    case [valeur19]: [liste d'instructions] break ;
    .
    case [valeurN] : [liste d'instructions] break ;
    default : [liste d'instruction] break ;
}
```

- "expression" : une expression du langage (variable, expression arithmétique, appel de fonction)
- "case [valeur1]" : "littéralement" cas où l'expression est égale à la valeur valeur1. Celle-ci doit avoir un type correspondant à celui de l'expression testée.
- "liste d'instructions" est une liste d'instructions séparées par des points virgule et qui se termine par un point virgule.
- "default" : représente le cas par défaut (est sélectionné quelque soit le résultat de l'expression.)
- "break" : est optionnel, s'il n'est pas présent, le switch continue à la ligne suivante (et donc exécutant d'autres branches pouvant s'appliquer, notamment la branche par défaut.)

On peut associer une même liste d'instructions à plusieurs cas (cf. valeur18 et valeur19).

### 2.1 Exercice 3

*Ecrire un programme qui lit un caractère en entrée et affiche :*

- "CCI" si la touche 'm' a été pressée
- "Principes des langages de programmation" si la touche 'c' a été pressée
- "2007" si la touche 'a' a été pressée
- "308" si la touche 's' a été pressée

### 2.2 Exercice 4

*Réécrire le programme précédant en acceptant les mêmes lettres mais aussi bien en majuscule qu'en minuscule. Ajouter un cas par défaut qui indique si la touche pressée n'est pas la bonne et quitte.*

## Deuxième partie

# Pour approfondir ...

### Exercice 5 : Ordre croissant ou décroissant

*Demander trois nombres à l'utilisateur puis s'il veut les classer en ordre croissant ou décroissant et affichez-les.*

### Exercice 6 : Equations du second degré

*Ecrire un programme qui demande trois nombres  $a, b, c$  de type double, et affichez ensuite la solution de l'équation du second degré*

$$ax^2 + bx + c = 0 \quad (1)$$

*Vous devrez inclure la librairie `math.h` et utiliser la fonction `sqrt(x)`.*

### Exercice 7 : Problème du Monnayeur

Le problème consiste à rendre la monnaie sur une somme, à l'aide de billets de 5 euros, de pièces de 1 et 2 euros, de pièces de 50 centimes, 20 centimes, 10 et 1 centime d'euros, de façon à minimiser le nombre de pièces rendues sachant que l'on connaît la somme due et la somme donnée par le client. On suppose que les sommes sont données en centimes d'euros et qu'il n'y a pas de pénurie de pièces ni de billets.

Exemple :

Prix à payer en centimes d'euros : 64

Somme versée par le client en centimes d'euros : 500

Il faut rendre :

2 pièces de 2 euros,

1 pièces de 20 centimes,

1 pièce de 10 centimes,

6 pièces de 1 centime.

*Ecrire un programme qui permet de résoudre ce problème et qui possède un affichage semblable à l'exemple.*

### Exercice 8 : Menu

*En utilisant `switch`, écrivez un programme avec un petit menu proposant la conversion des angles du TD précédant, les exercices 5, 6 et 7 de ce TD. Votre menu devra également proposer de quitter directement et indiquer si la touche pressée n'est pas dans la liste.*