

# C 3 : Boucles définies

Juliusz Chroboczek

28 septembre 2022

## 1 Boucles définies

Une *boucle* est une structure de contrôle qui sert à exécuter le même bloc de code de multiples fois. Une *boucle définie*, ou *boucle for*, est une boucle dont l'exécution est contrôlée par un *compteur de boucle* dont la valeur varie entre deux valeurs qui sont déjà connues avant d'entrer dans la boucle.

**Syntaxe et sémantique des boucles définies** Le fragment de code suivant exécute les deux instructions `printf` pour toutes les valeurs de `i` allant de 1 à 10 :

```
int i;
for(i = 1; i <= 10; i = i + 1) {
    printf("J'ai collé %d timbres.\n", i);
    printf("Il m'en reste %d.\n", 10 - i);
}
```

Une boucle `for` est composée du mot-clé `for` suivi de trois expressions entre parenthèses, séparées par des points-virgule, puis d'un bloc (des déclarations et des instructions entre accolades) :

$$\text{for}(e_1; e_2; e_3) \text{ bloc}$$

L'exécution de la boucle procède de la façon suivante :

1. l'expression  $e_1$ , dite *initialisation*, est exécutée;
2. le *test*  $e_2$  est évalué; s'il est faux, on sort de la boucle, et l'exécution de la boucle est terminée;
3. le corps de la boucle est exécuté;
4. l'expression  $e_3$  est exécutée;
5. on recommence à l'étape 2.

**Pas arbitraires** L'instruction  $e_3$  d'une boucle définie sert typiquement à incrémenter le compteur de boucle, c'est-à-dire à ajouter une constante à sa valeur. Cette constante n'est pas forcément égale à 1. Par exemple, le fragment de code suivant :

```
int i;
for(i = 0; i < 10; i = i + 1)
    printf("%d est pair.\n", 2 * i);
```

peut aussi s'écrire :

```
int i;
for(i = 0; i < 20; i = i + 2)
    printf("%d est pair.\n", i);
```

Lorsque le pas est négatif, il faudra inverser le sens du test de terminaison :

```
int i;
for(i = 9; i >= 0; i = i - 1)
    printf("Il me reste %d timbres.\n", i);
```

## 2 Quelques abréviations

Utilisée toute seule,

- l'expression `i++` est équivalente à `i = i + 1`;
- l'expression `i--` est équivalente à `i = i - 1`;
- l'expression `i += c` est équivalente à `i = i + c`;
- l'expression `i -= c` est équivalente à `i = i - c`.

Ces abréviations sont souvent utilisées dans l'expression  $e_3$  d'une boucle définie.