

INF3135

Construction et maintenance de logiciels

Cours 2: Bases du C, partie 1

Alexandre Blondin Massé

Université du Québec à Montréal
Département d'informatique

Été 2020

Table des matières

1 Organisation du cours

2 Résumé des capsules

3 Exercices

Organisation du cours

Informations générales

- **Enseignant:** Alexandre Blondin Massé
- **Courriel:** blondin_masse.alexandre@uqam.ca
- **Site personnel:** <http://lacim.uqam.ca/~blondin>
- **Site du cours:** <http://lacim.uqam.ca/~blondin/fr/inf3135>
- **Plan de cours:** [cliquer ici](#)
- **Description des labos:** [dépôt GitLab](#)
- **Théorie:** [capsules vidéos](#) à visionner **avant** les cours
- **Matériel:** [groupe GitLab du cours](#)
- **Support en ligne:** [équipe Mattermost](#) plutôt que [canal Slack](#)

Modalités d'évaluation

Travaux pratiques (60%)

- **TP1 (seul)**: initiation au langage C (20%)
- **TP2 (seul)**: maintenance d'un programme (20%)
- **TP3 (seul ou équipe)**: construction d'un programme (20%)

Quiz (10% chacun, 40% au total)

- **Quiz 1**: 26 mai
- **Quiz 2**: 23 juin
- **Quiz 3**: 14 juillet
- **Quiz 4**: 11 août

Double seuil

Pas de double seuil

Résumé des capsules

Types

Numériques

- **Entiers:** bool, char, short, int, long, long long
- **Variante:** signed ou unsigned
- **Valeurs flottantes:** float, double, long double
- **Énumératifs:** enum (représentés par des entiers)

Types complexes

- Tableaux
- **Structures** (struct)
- **Unions** (union)
- **Vide:** void
- **Pointeurs:** plus tard
- **Synonymes:** avec typedef

Variables et constantes

Variables

- **Déclaration:** mémoire réservée
- **Initialisation** optionnelle
- Automatique (`auto`)
- Statique (`static`)
- Externe (`extern`)

Constantes

- Directive `#define`
- Mot réservé `const`
- Avec `enum`
- **Littéraux:** `u` (non signée), `l` (long), `o` (octal), `0x` (hexadécimal), `'c'` (caractère), `"..."` (chaînes)
- **Caractères spéciaux:** `\n`, `\t`, `\\`, `\'`, `\"`

Structures de contrôle

Répétitives

- `for`
- `while`
- `do/while`
- `break/continue`

Conditionnelles

- `if/else if/else`
- `switch/case/break/default`

Opérateurs (1/2)

Arithmétiques

+, -, *, /, %

Comparaison

==, !=, >, >=, <, <=

Logiques

!, &&, ||

Affectation

=, +=, -=, *=, /=, %=, ++, --

Opérateurs (2/2)

Séquençage

, (rarement utilisé)

Ternaire

? :

Bit à bit

&, |, ^

Taille en octets

sizeof

Conversions

Implicites

- Entre valeurs **entières**:

`bool → char → short → int → long → long long`

- Entre valeurs **flottantes**:

`float → double → long double`

- Promotion automatique **entier** → **flottant**

Explicites

`(type)variable`

Priorité des opérateurs

Arité	Associativité	Par priorité décroissante
2	→	(), []
2	→	->, .
1	←	!, ++, --, +, -, (int), *, &, sizeof
2	→	*, /, %
2	→	+, -
2	→	<, <=, >, >=
2	→	==, !=
2	→	&&
2	→	
3	→	? :
1	←	=, +=, -=, *=, /=, %=
2	→	,

Exercices

Exercices

- Écrire une fonction

```
void print_n(char c, unsigned int n);
```

qui affiche n fois le caractère c sur stdout

- Écrire une fonction

```
void print_triangle(char c, unsigned int n);
```

qui affiche n fois le caractère c sur une première ligne, suivi de $n - 1$ fois le même caractère sur une deuxième ligne, ainsi de suite, jusqu'à la dernière ligne qui contient 1 fois le caractère c sur stdout

- Écrire une fonction

```
void afficher_damier(char c, unsigned int m, unsigned int n);
```

qui affiche un damier (c alterné avec espace) de m lignes et n colonnes sur stdout