

Algorithmique et programmation en C

Les structures de base

Auteur du document: Marwa HADDAD-ISSTE borj cedria



Ce document est disponible sous licence creative commons (CC BY-NC-SA 3.0)

REL de références :

Url ref1 :

http://ressources.unisciel.fr/algoprogram/s01alprg/emodules/al00macours1/co/al00macours1_web.html

Licence creative commons de type 3: (CC BY-

NC-SA 3.0)



Karine Zampieri, Stéphane Rivière, Béatrice Amerein-Soltner- **L'Algorithmique [al]**

version 28 décembre 2016

<http://ressources.unisciel.fr> License: Creative Commons BY-NC-SA.

Url ref2 :

[https://www.oercommons.org/courses/advanced-algorithms-fall-](https://www.oercommons.org/courses/advanced-algorithms-fall-2008/view)

[2008/viewhttps://www.oercommons.org/courses/advanced-algorithms-fall-2008/view](https://www.oercommons.org/courses/advanced-algorithms-fall-2008/view)

Creative Commons Attribution-NonCommercial-ShareAlike 3.0



Erik Demaine, and Srinivas Devadas. *6.006 Introduction to Algorithms*. Fall 2011. Massachusetts Institute of Technology: MIT OpenCourseWare, <https://ocw.mit.edu>. License: Creative Commons BY-NC-SA.

Le langage

Séquence de lettres (A...Z, a...z) avec ou sans accents, de chiffres (0...9) ou du caractère souligné (_). Il doit commencer par une lettre ou un souligné.

Variables, types et valeurs

Variable

Élément informatique ' qu'un algorithme peut manipuler. Décrite par :

- Un nom ou identifiant unique qui la désigne.
- Un type qui définit de quel « genre » est l'information associée.
- Une valeur qui peut évoluer au cours de l'algorithme mais qui doit respecter le type.

Types intégrés

- Entier : Pour les entiers relatifs Z

ENC	Il y a cinq types de variables entières (« integer » en anglais) : char ; short int, ou plus simplement short ; int ; long int, ou long ; long long int, ou long long (ce type a été ajouté depuis la norme C99). Comme évoqué en introduction, le type caractère char est particulier, et sera étudié en détail plus bas. Les types entiers peuvent prendre les modificateurs signed et unsigned qui permettent respectivement d'obtenir un type signé ou non signé. Ces modificateurs ne changent pas la taille des types. Le langage ne définit pas exactement leurs tailles, mais définit un domaine de valeurs minimal pour chacun.
-----	---

- Réel : Pour les nombres réels (approchés) R

ENC	Les nombres réels ne pouvant tous être représentés, sont approximés par des nombres à virgule flottante. Comme dans le cas des entiers, il existe plusieurs types de nombre à virgule flottante. En voici la liste triée par précision croissante : float ; double ; long double
-----	--

- Booléen : Le domaine B des booléens (vrai / faux)

ENC	Le langage (jusqu'à la norme C99) ne fournit pas de type booléen. La valeur entière 0 prend la valeur de vérité faux et toutes les autres valeurs entières prennent la valeur de vérité vrai. La norme C99 a introduit le type _Bool, qui peut contenir les valeurs 0 et 1. Elle a aussi ajouté l'en-tête , qui définit le type bool qui est un raccourci pour _Bool, et les valeurs true et false[12] . Nota : toute expression utilisant des opérateurs booléens (voir opérateurs), retourne 1 si l'expression est vraie et 0 si elle est fausse, ce qui rend quasiment inutile l'usage du type booléen.
-----	---

- Caractère : Le domaine A des caractères (alphanumériques et ponctuations)

ENC	À l'origine, le type permettant de représenter un caractère est char. Même si un char n'est plus toujours suffisant aujourd'hui pour représenter un caractère quelconque. Ce type est un peu plus particulier que les autres, d'une part parce que sa taille définit l'unité de calcul pour les quantités de mémoire (et donc pour les tailles des autres types du langage) et d'autre part son domaine de valeur peut grandement varier de manière relativement inattendue. Par définition, la taille du type char, notée sizeof(char), vaut toujours 1. Cependant, il faut faire attention : contrairement à ce qu'on pense souvent, un char au sens du C ne vaut pas toujours un octet. Il occupera au minimum 8 bits, mais il existe des
-----	---

	architectures, relativement spécialisées il est vrai, ayant des char de 9 bits, de 16 bits, voire plus. Même si, dans une large majorité des cas, les compilateurs utilisent des char de 8 bits, à la fois par simplicité (les machines modernes fonctionnent généralement en 8, 16, 32 ou 64 bits) et pour éviter des problèmes de portabilité de code (beaucoup de codes C existants reposent sur l'hypothèse que les char font 8 bits, et risqueraient de ne pas marcher sur une autre architecture)
--	---

- Chaîne : Le domaine T des textes (suite de caractères)

EN C	Une chaîne de caractère, comme son nom l'indique, est une suite de caractères avec la particularité d'avoir un caractère nul (0) à la fin. Une chaîne de caractère est en fait implémentée en C avec un tableau de type char. Ce chapitre vous donne les rudiments sur les chaînes de caractère pour une première approche. Pour en savoir plus sur les rares fonctions de chaînes de caractères
------	--

Littéraux

Entier : Suite de chiffres éventuellement préfixé par un signe (+ ou -). • Réel : S'écrit en notation décimale ou en notation scientifique. • Booléen : Identifie le Vrai et le Faux. • Caractère : Se place entre quotes ('). • Chaîne : Se place entre guillemets (").

Déclarations

Déclaration de variables

```
Variable nomVar : TypeVar
Variable nomVar1, nomVar2, ... : TypeVar
```

Cette ligne déclare les variables var1, var2, ..., varN de type T. Une variable est dite locale, si elle est définie à l'intérieur d'une fonction et globale si définie en-dehors. Par exemple en C:

```
int jour; /* 'jour' est une variable globale, de type entier */

int main(void)
{
    double prix; /* 'prix' est une variable locale à la fonction 'main', de type réel */
    return 0;
}
```

Définition de constante

```
Constante nomConst <- expression
```

Structure générale d'un algorithme

```
Début
| instruction1
| instruction2
| ...
Fin
```

EN C	<p>Les instructions se terminent par un point-virgule (;), on peut placer autant d'instructions que l'on veut sur une même ligne (même si ce n'est pas conseillé pour la lisibilité du code). Les blocs d'instructions commencent par une accolade ouvrante ({) et se terminent par une accolade fermante (}). Les instructions doivent obligatoirement être déclarées dans une fonction : il est impossible d'appeler une fonction pour initialiser une variable globale par exemple (contrairement au C++).</p> <pre> /* une instruction */ i = 1; /* plusieurs instructions sur la même ligne */ i = 1; j = 2; printf("bonjour\n"); /* un bloc */ { int i; i = 5; } /* l'instruction vide */ ; </pre>
------	---

```

Algorithme nomAlgo
| déclaration_des_variables_et_constants
Début
| saisie_des_données
| instructions_utilisant_les_données_lues
| communication_des_résultats
Fin

```

Saisie de données

Saisir(nomVar1, nomVar2, ..., nomVarN)

Affichage de résultats

AfficherSS(expr1,expr2,...,exprN) // SANS retour de ligne

Afficher(expr1,expr2,...,exprN) // AVEC retour de ligne

Affectation interne

nomVar <- expression

Expressions algébriques

Expression, opérandes, opérateurs

Eventuellement accompagnés de parenthèses, une ´ expression est une séquence « bien formée » (au sens de la syntaxe) d’opérandes (valeurs littérales, variables ou expressions) et d’opérateurs destinée à l’évaluation.

Opérateurs arithmétiques

Opérateur Mathématique	Signification	Équivalent Algorithmique
+	(unaire) valeur	$+a$
-	(unaire) opposé	$-a$
+	addition	$a + b$
-	soustraction	$a - b$
*	multiplication	$a * b$
/	division décimale	a / b
div	division entière	$\text{DivEnt}(a,b)$
mod	modulo (reste de la division entière)	$\text{Modulo}(a,b)$
^	élévation à la puissance	$a ^ b$

Ordre de priorité des opérateurs arithmétiques

Comme en mathématique : 1. Les opérateurs unaires (+, -) (priorité la plus élevée) Structures de base [bs] - Résumé de cours 4 2. L'opérateur d'exponentiation (^) (s'il existe) 3. Les opérateurs multiplicatifs (*, /, div, mod) 4. Les opérateurs additifs (+, -) (priorité la plus basse) La règle d'associativité s'applique en cas d'ambiguïté entre opérateurs du même ordre de priorité.