



TP VVL - TESTS UNITAIRES

CODE39

---

# CODE39

## DEFINITION

Wikipedia:

“Le Code 39 (aussi appelé par abus Code 3 sur 9 et parfois Alpha39) est une symbologie code-barres, utilisée, entre autres, pour le marquage des médicaments en pharmacie, en France et dans certains autres pays européens (Allemagne, Italie, ...). Il a été utilisé jusqu'à récemment dans le secteur automobile, constructeurs et équipementiers, en rapport avec les normes internationales Odette-GALIA1, qui utilisent maintenant les symbologies Code 128 et PDF-4172”

## CODAGE

Le codage de chaque caractère est la succession de 9 bits dont exactement 3 sont à 1 (barre large).

La liste des 43 caractères utilisables est la suivante:

- Ensemble des caractères alphabétiques en majuscule: **[A-Z]**
- Ensemble des caractères numériques **[0-9]**
- Les caractères spéciaux suivants: **{ espace, -, \$, %, ., /, + }**

Le caractère “\*” sert de délimiteur de début et de fin. Un Code 39 commence et finit toujours par le caractère « \* » qui sert de déclencheur pour le lecteur de code-barres.

Son plus grand avantage est que comme il ne requiert aucune somme de contrôle, il peut être facilement intégré à tout système d'impression via une police de caractères spéciale.

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z		
0	1	2	3	4		*
5	6	7	8	9		
(SPACE)	-	\$	%	.	/	+

A tout moment vous pouvez générer le code-barres sur la base d'une chaîne de caractères en utilisant une police de caractère comme celle-ci: <http://www.dafont.com/fr/code39.font>

Ou encore encoder votre chaîne de caractères à l'aide d'outil en ligne comme celui-ci:

<https://racoindustries.com/barcodegenerator/1d/code-39/>

Vous pouvez également le relire (même sur un écran) en utilisant votre smartphone est votre application “barcode scanner” préférée.

## CUNIT

À l'origine, Kent Beck crée l'environnement de test sUnit pour le langage Smalltalk en octobre 1994. En 1997, Kent Beck rencontre Erich Gamma avec lequel il crée JUnit qui, suite à sa popularité, entraînera la création de nombreux frameworks de tests unitaires, cet ensemble se nomme xUnit.

Le terme générique « xUnit » désigne un outil permettant de réaliser des tests unitaires dans un langage donné (dont l'initiale remplace « x » le plus souvent) : CppUnit pour le C++, CUnit pour le C, PHPUnit pour PHP, JUnit pour Java, etc ...

CUnit permet d'intégrer facilement des tests unitaires au sein du programme en C, d'effectuer des tests de validation et de vérification.

La documentation de CUnit est disponible ici: <http://cunit.sourceforge.net/doc/index.html>

## UTILISATION

Afin de pouvoir exécuter au mieux un ensemble de tests, il est nécessaire de réaliser les différentes étapes suivantes:

- Ecrire les tests
- Initialiser le catalogue
- Ajouter les suites de tests dans le catalogue
- Ajouter les tests dans les suites de tests
- Exécuter les tests
- Terminer proprement l'exécution des tests

De plus CUnit permet l'utilisation de 4 modes:

- Automated: non-interactif, sortie dans fichier XML
- Basic: non-interactif avec sortie standard
- Console: interactif avec utilisation de la console
- Curses: interactif avec utilisation de NCurses (Linux+Unix only)

## NON REGRESSION ET REFACTORING

Après chaque test effectué, à partir du rapport de test il conviendra de réviser la structure du code tout en garantissant une non-regression (à l'aide de ces tests).

---

## A. TESTS UNITAIRES

A l'aide du framework CUnit, l'étudiant mettra en oeuvre des procédures de tests unitaires en adéquation avec le sujet. L'objectif est de réaliser un ensemble de fonctions qui vont permettre de valider une chaîne de caractère avant son encodage en Code39.

### 1. ENCODAGE CODE39

- Ecrire le corps du programme en C qui permette la saisie d'une chaîne de caractère sur la console.
- Ecrire une fonction qui compose la chaîne de caractère prête à être encodée en Code39

### 2. VÉRIFICATION DU CODE

- Ecrire une fonction de tests de cette chaîne de caractère suivant la norme énoncée plus haut.
- Intégrer des tests unitaires au programme en utilisant CUnit avec le mode Basic dans un premier temps puis Automated.
- Mettre en oeuvre le mode interactif des tests en utilisant le mode "Console".

## B. TEST DRIVEN DEVELOPMENT

Le Test Driven Development (TDD), issu de la méthodologie eXtreme Programming, appliqué au développement logiciel permet de tester un code source par construction. Il invite le programmeur à écrire les tests unitaires avant le programme.

1. La séquence préconisée est la suivante
2. Ecriture d'un test
3. Vérifier qu'il échoue (le code source n'existe pas)
4. Ecrire le code source minimal et nécessaire pour que le test passe
5. Code refactoring: l'améliorer tout en garantissant une non régression

### 1. CODE39 AMELIORE: UTILISATION D'UNE VALEUR DE CONTROLE

La norme Code 39 ne possède pas de valeur de contrôle (checksum). Une extension de cette norme d'encodage de barcode permettant d'améliorer la garantie d'intégrité du barcode est appelée *modulo 43*.

Comme son nom le suggère, il s'agit d'insérer en fin de chaîne de caractère un caractère de contrôle correspondant au modulo 43 de la somme des valeurs de la chaîne de caractères.

La valeur de chaque caractères à encoder est dans la table de correspondance suivante:

---

0	1	2	3	4	5	6	7	8	9	A
0	1	2	3	4	5	6	7	8	9	10
B	C	D	E	F	G	H	I	J	K	L
11	12	13	14	15	16	17	18	19	20	21
M	N	O	P	Q	R	S	T	U	V	W
22	23	24	25	26	27	28	29	30	31	32
X	Y	Z	-	.	Espace	\$	/	+	%	
33	34	35	36	37	38	39	40	41	42	

- Ecrire, en TDD, les fonctions nécessaires à la génération du caractère Modulo 43 supplémentaire
- Ecrire en TDD, la fonction nécessaire à l'ajout de ce caractère en bout de chaîne
- Réaliser un test de non régression et réaliser un code refactoring si nécessaire
- Utiliser successivement un mode non interactif puis interactif de CUnit pour réaliser un rapport de test final