TPs: Opérateurs Binaires

Van Oudenhove Didier

10 novembre 2023

Première partie

Exercices sur les opérateurs binaires

Pour rappels les opérateurs binaires sont les opérateurs AND, OR, XOR et NOT ainsi que les opérateurs de décalage gauche et droit. Cf Chapitre 6.6 dans le syllabus.

1 Réalisez une fonction qui indique si une lettre est une majuscule

Écrivez la fonction « booléen estMajuscule(char lettre) » qui devra en fonction de la variable « lettre » renvoyer « vrai » si c'est une majuscule sinon « faux » si c'est une minuscule.

Utilisez le fait que le bit de position 5 (2^{32}) est à 0 pour une majuscule et à un pour une minuscule.

'A'	0	1	0	0	0	0	0	1
'a'	0	1	1	0	0	0	0	1

Implémentation

Après avoir écrit le pseudo-code, implémentez le dans les 2 langages:

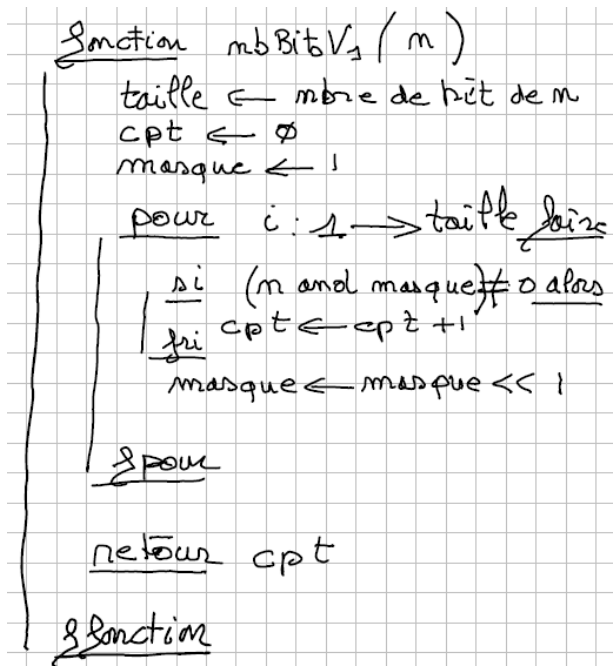
Java: Réalisez la fonction en Java dans votre classe « MyMath » et testez la avec un test unitaire dans « TestMyMath ». En *Java* pour obtenir le code du caractère vous devez faire « (int) lettre ».

Python Réalisez la fonction en Python dans le module « mymath » et le test unitaire dans le module « test_mymath ». En *Python,* pour obtenir le code du caractère vous devez faire « ord(lettre) ».

2 Calculez le nombre de bits à un dans une variable entière

En classe nous avons réalisé cet exercice en utilisant une boucle « pour », qui décalait le masque d'une position vers la gauche à chaque passage dans la boucle. Voici le pseudo-code que nous avions généré:

Algorithme 1 nbBitsV1



Hyp: n est un nombre entier (signé ou non signé)

Fonction: nbBitsV1(n)

- IN: n le nombre à analyser
- OUT: le nombre de bits à 1

Variables:

- cpt: un entier; le compteur de bits à 1
- masque: un entier; le masque
- taille: le nombre de bits de la variable n

2.1 Version 2 à faire: nbBitsV2(n)

On vous demande ici de réaliser un autre algorithme plus performant qui va utiliser une boucle « Tant Que » . Cette boucle devra s'arrêter lorsque les bits restants ne contiennent plus que des zéros ou que vous ayez parcouru tous les bits de la variable « n ». Dans cet algorithme, je vous demande de ne pas modifier « n »!

Prenons l'exemple suivant:

n:	0	0	0	0	0	0	0	1	==>	n:	0	0	0	0	0	0	0	1
masque	0	0	0	0	0	0	0	1		masque	0	0	0	0	0	0	1	0

On doit pouvoir s'arrêter ici!

Implémentation

Après avoir écrit le pseudo-code, implémentez le dans les 2 langages:

Java: Réalisez la fonction en Java dans votre classe « MyMath » et testez la avec un test unitaire dans la classe « TestMyMath ».

Python L'équivalent en Python sera peut-être un peu plus technique étant donné sa manière à coder les entiers, mais vous pouvez également le faire, ainsi que son test unitaire.

2.2 Version 3 à faire: nbBitsV3(n)

Imaginez encore une autre version d'algorithme où je vous autorise à modifier la valeur de « n ». Il faut aussi s'arrêter dès que possible et donc avoir une boucle « Tant Que »!

Indice: dans cette version le masque sera fixe et on décalera le contenu de la variable n.

Implémentation

Après avoir écrit le pseudo-code, implémentez le dans les 2 langages:

JAVA: Réalisez la fonction en Java dans votre classe « MyMath » et testez la avec un test unitaire dans « TestMyMath ».

Python L'équivalent en Python sera peut-être un peu plus technique étant donné sa manière à coder les entiers, mais vous pouvez également le faire ainsi que son test unitaire.

3 Conversion en Hexadécimal

L'hexadécimal correspond à la base 16, elle est très utilisée pour compresser du binaire, en effet elle permet de diviser la taille du nombre par 4:

Exemple:

base 2	0	1	0	1	1	0	1	1
base 16	5				B			

Chaque groupe de 4 bits (en commençant par la droite) peut être remplacé par un chiffre hexa [0..9A..F]
0000_2 à 1001_2sont directement associé à 0_{10} à 9_{10} alors que les codes suivants sont associés à une lettre hexa:

Code binaire sur 4 bits	Lettre Hexadécimale correspondante
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

3.1 Convertir un nombre entier en sa chaîne de caractères hexadécimale

Fonction: « String convertToHex(int n) »

In: « n » un nombre entier sur 32 bits

Out: une chaîne de caractères qui commence par "0x....."

Exemple:

Si n= 3914 alors votre fonction devra fournir la chaîne "0xF4A"

Écrivez le pseudo-code

En pseudo-code vous pouvez utiliser le + pour concaténer:

$n \leftarrow$ "$0x$"	
$n \leftarrow n +$ "F"	n contiendra "OxF"

Implémentation du code en Java pour une variable de 32bits

Réalisez la fonction en Java dans votre classe « MyMath » et testez la avec un test unitaire dans « TestMyMath ». Comme pour le pseudo-code, l'opérateur « + » effectue une concaténation.

4 Encodage en UTF8

UNICODE_16 est un encodage des caractères sur 2 bytes mais vous avez certainement entendu parlé de l'encodage UTF8, il s'agit d'une compression de l'UNICODE qui fonctionne de la manière suivante:

Code UNICODE	nombre de bits utilisés		Nbr de bytes en UTF8	Codification en UTF8
	min	max		
0000 0000 0XXX XXXX	0	7	1	0XXX XXXX
0000 0XXX XXXX XXXX	8	11	2	110X XXXX 10XX XXXX
XXXX XXXX XXXX XXXX	12	16	3	1110 XXXX 10XX XXXX 10XX XXXX

Exemple 1:
Soit la lettre 'A', son code est « 65 » sur 16 bits soit | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |,
il nécessite donc 7 bits, on est dans le premier cas, l'encodage en UTF8 utilisera 1 byte et sera:

0	1	0	0	0	0	0	1

Exemple 2:
Le code Unicode de la lettre « à » est « 224 » soit | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |,
il nécessite donc 8 bits, on est dans le deuxième cas, l'encodage en UTF8 utilisera 2 bytes et sera:

1	1	0	0	0	0	1	1	1	0	1	0	0	0	0	0

4.1 Réalisez une fonction qui indique le nombre de bytes nécessaires pour coder un caractère:

Fonction: « int nbBytesUtf8(char c) »

In: « c » un caractère

Out: un entier qui sera égal à 1, 2 ou 3

Écrivez le pseudo-code

En pseudo-code, vous pouvez écrire: n<— code Unicode du caractère c

Implémentation

Après avoir écrit le pseudo-code, implémentez le dans les 2 langages:

Java: Réalisez la fonction en Java dans votre classe « MyMath » et testez la avec un test unitaire dans « TestMyMath ». Pour obtenir le code du caractère vous devez faire « (int) c ».

Python Faite de même en Python. Pour obtenir le code du caractère vous devez faire « ord(c) ».

4.2 Réalisez une fonction qui va décoder un code UTF8 en UNICODE

Fonction: « int decodeUtf8(int utf8) »

In: vous recevez en entrée un entier qui correspond à un code Utf8

Out: votre fonction doit renvoyer le code Unicode associé.

Exemple: decodeUtf8(0b1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0) ==> devra donner « 224 »

Écrivez le pseudo-code

Implémentation

Après avoir écrit le pseudo-code, implémentez le dans les 2 langages:

Java: Réalisez la fonction en Java dans votre classe « MyMath » et testez la avec un test unitaire dans « TestMyMath ».

Python Faite de même en Python.