Les opérations

Jean-Christophe Dubacq

S1 2016

1 Les opérations

1.1 Les entiers

1.1.1 De la fonction à l'algorithme

La numération grecque (simple) est proche de la numération romaine que vous connaissez : on note les nombres comme suit :

1	5	10	50	100	500	1 000	5 000	10 000	50 000
I	Γ	Δ	Γ_Δ	H	Γ_H	X	Γ_X	M	Γ_M

C'est à la différence près que l'on a pas de règle soustractive : le nombre 4 s'écrit IIII, pas IΓ. La position des chiffres n'a théoriquement aucune importance, mais on les classait dans l'ordre décroissant de valeur.

Q1 Ce système est-il un système de numération positionnelle ?

Q2 Écrivez votre âge et votre date de naissance en numération grecque.

Q3 Écrivez un algorithme d'addition des nombres représentés en numération grecque. Est-ce que cet algorithme est le même qu'en décimal ?

Q4 Faites l'addition de votre âge et de votre année de naissance avec votre algorithme (vous devriez obtenir XXΔIIII ou XXΔIII). De quelle représentations partez-vous ?

Q5 Faites la même chose en décimal. De quelles représentations partez-vous ? Est-ce que l'algorithme est le même ? Est-ce que la fonction calculée est la même ?

1.1.2 Calcul en binaire et hexadécimal

Q6 Faites les additions en binaire : `0b1101 0101`+`0b1110 0101` ; `0b1,1`+`0b110`+`0b100,1`+`0b111,1`+`0b1010,1`+`0b100,1`.

Q7 Faites les opérations suivantes en hexadécimal : $0x122 + 0x233$; $0x87 + 0x54$; $0x18 + 0x9$; $0xED + 0xED$; $0x100 - 0x3$.

Q8 Faites la multiplication suivante : 17×129 à la fois en décimal et en binaire.

Q9 Faites la multiplication suivante en binaire : 110110×1101.

1.2 Addition et codage

1.2.1 Limites de la multiplication

Expliquez pourquoi le résultat d'une multiplication de deux nombres représentés dans l'un des 4 codes classiques est toujours représentable à condition de doubler la taille du code.

1.2.2 Addition en C2

Q10 Faites les opérations suivantes en transformant les nombres au préalable en codage C2 sur 8 bits (résultat aussi en C2 sur 8 bits) :

- 45+17
- 45-17 (soit 45+(-17))
- -17-17
- 17-45
- 221+45

Dites aussi si le résultat obtenu est correct et s'il est représentable.

1.3 Les champs de bits

1.3.1 Tables de vérité

Q11 Faites une table qui montre toutes les paires d'arguments possibles pour les opérateurs AND, OR, XOR et qui montre le résultat à côté.

A	B	A×B
0	0	
0	1	
1	0	
1	1	

A	B	A+B
0	0	
0	1	
1	0	
1	1	

A	B	A⊕B
0	0	
0	1	
1	0	
1	1	

1.3.2 Opérations booléennes

Q12 Que vaut 0b10000110 AND 0b11101001 ?

Q13 Que vaut 0b10000110 OR 0b11101001 ?

Q14 Que vaut 0b10000110 XOR 0b11101001 ?

Q15 Que se passe-t-il si on calcule (a est une variable booléenne) : $a + 0$? $a + 1$? $a \times 0$? $a \times 1$? $a + a$? $a + a + a + a + a + a$?

Q16 Démontrez que $a + ab = a$;

Q17 Démontrez que $a + bc = (a + b)(a + c)$;

Q18 Démontrez que $a + \overline{a}b = a + b$;

1.3.3 Analyse d'un masquage

Dans un champ de bits qui contient $a = 0b11001001$, on veut faire les choses suivantes :

Q19 On veut vérifier si le bit 0 est actif ou non. Décomposez l'opération.

Q20 On veut changer le bit 1 en 1 et le bit 3 en 0. Décomposez les opérations qui permettent de le faire.

Q21 Changez le bit 5, en expliquant les valeurs intermédiaires.

1.3.4 Analyse de touches

Dans un système, la fonction `keyEvent()` renvoie une valeur entière sur 16 bits (dont 5 ignorés) :

— Les 8 premiers bits correspondent au numéro de la touche sur le clavier (pour les touches ordinaires)
— Le 9e bit correspond à la touche SHIFT (1 : pressée, 0 : pas pressée)
— Le 10e bit correspond à la touche CONTROL (1 : pressée, 0 : pas pressée)
— Le 11e bit correspond au fait d'appuyer sur une touche (1) ou de l'avoir juste relachée (0)

Q22 Écrivez un programme qui appelle cette fonction (`a=keyEvent()`) puis qui en fonction de `a` affiche un texte du genre : « Vous venez de lâcher la touche 27 en ayant SHIFT appuyé et CONTROL lâché »