

Codages d'information, système de numération

M. Dubacq

S1D 2009

1 Préparatifs

Objectif : *Connaître les puissances de 2 et les bases de la numération hexadécimale.*

Écrivez la liste de toutes les puissances de 2, de 2^{-4} à 2^{16} . Écrivez une table de conversion des chiffres hexadécimaux et octaux vers le codage natuel écrit en binaire (4 bits ou 3 bits).

2 De la fonction à l'algorithme

Objectif : *Maîtriser la différence entre fonction et algorithme. Concevoir un algorithme simple. Examiner l'influence de la représentation sur l'algorithme utilisé pour une même fonction.*

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	X	Γ _X	H	Γ _H	M	Γ _M

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

1. Ce système est-il un système de numération positionnelle ?
2. Écrivez votre âge et votre date de naissance en numération grecque.
3. Écrivez un algorithme d'addition des nombres représentés en numération grecque. Est-ce que cet algorithme est le même qu'en décimal ?
4. Faites l'addition de votre âge et de votre année de naissance avec votre algorithme (vous devriez obtenir HHΓIII ou HHΓIIII). De quelle représentations partez-vous ?
5. 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 ?

3 Quantité d'information

Objectif : *Comprendre la nature logarithmique de la mesure en bits.*

1. Dans votre classe de TD, jusqu'à combien devez-vous aller si vous donnez un numéro unique à chaque étudiant en partant de 0 ? Écrivez ces nombres en binaire. Combien de bits d'information sont nécessaires pour désigner un élève parmi tous les autres ?
2. S'il y a un élève de plus, combien de bits supplémentaires seraient nécessaires ?
3. Combien d'élèves sont nécessaires pour qu'on ait besoin de 6 bits ? Et 7 bits ? et 10 bits ?

4 Conversions

Objectif : *Comprendre la différence entre quantité d'information et longueur de codage, savoir faire des conversions bits \leftrightarrow octets, savoir faire des multiplications sans calculatrice, comprendre la différence entre l'échelle binaire et l'échelle décimale pour les unités.*

1. En Syldavie, les plaques d'immatriculation comportent 3 chiffres (décimaux) et une lettre (l'alphabet syldave comporte 32 lettres). Combien de plaques différentes peuvent exister ? Combien de bits d'information représente une voiture (parmi les autres) ?
2. Combien de bits faut-il pour coder un chiffre décimal ? Une lettre syldave ?
3. Si on met bout à bout les codages de trois chiffres décimaux et d'une lettre syldave, quelle est la longueur du codage d'une voiture syldave ? Comment expliquez-vous la différence avec la réponse à la question 1 ?
4. Le nouveau ministre des transports syldave décide de changer les plaques pour utiliser trois lettres. Est-ce que c'est possible, sachant que toutes les plaques sont utilisées ?
5. Convertissez 24×10^8 bits en Go.

6. Convertissez 2^{16} octets en *Mib*. Donnez une approximation en *Mb*. Quel est l'ordre de grandeur de l'approximation faite ?
7. Un élément d'ordinateur est capable d'émettre 1024 bits en 0,5 nanosecondes. Quel est le débit (quantité d'information divisée par le temps) de cet élément en bits par secondes ? Quelle est la bonne unité pour ce débit ?

5 Conversion en binaire et hexadécimal

Objectif : *Savoir convertir des nombres de binaire/hexadécimal en décimal et inversement, maîtriser l'écriture des réels dans d'autres bases, comprendre la notion de poids des chiffres dans un système positionnel*

1. Écrivez en binaire et en hexadécimal les nombres décimaux suivants : 28 ; 149 ; 1285 ; 0,3125 ; 164,3125.
2. Convertissez en décimal les nombres suivants : 0x48 ; 0xA1C ; 0b1010010010011111 ; 0b1010,0011.
3. Comment trouver midi à quatorze heures ?

6 Calcul en binaire et hexadécimal

Objectif : *Maîtriser les conversions et les additions en base 2 et 16, maîtriser la multiplication en base 2, comprendre la notion de poids des chiffres dans un système positionnel*

1. Faites les additions en binaire : 0b1101 0101+0b1110 0101 ; 0b1,1+0b110+0b100,1+0b111,1+0b1010,1+0b100,1.
2. Faites les opérations suivantes en hexadécimal : 0x122+0x233, 0x87+0x54, 0x18+0x9, 2×0xED, 0x100−0x3.
3. Faites la multiplication suivante : 17 × 129 à la fois en décimal et en binaire.
4. Faites la multiplication suivante en binaire : 110110 × 1101.

7 Codage des entiers

Objectif : *Comprendre la différence entre codage et écriture d'un nombre, maîtriser les quatre codages classiques des entiers, comprendre les limites des codages*

Ce tableau comporte des cases inutilisées (la première ligne est un exemple). Complétez-le :

Décimal	Écriture Binaire	Type de codage	Codage (binaire)	Codage (hexa)
-18	-1 0010	VA+S (8 bits)	1001 0010	0x92
424		NAT (16 bits)		
-138		C2 (16 bits)		
	-111 0011	C1 (8 bits)		
-4197		VA+S (24 bits)		
-84				0xAB
341		NAT (8 bits)		

8 Codage des nombres flottants

Objectif : *Comprendre la notation scientifique, le procédé de codage par concaténation de champs, la conversion binaire/hexadécimal*

Ce tableau comporte des cases inutilisées. Complétez-le :

Décimal	Binaire	Virgule flottante	E	Codage IEEE754			Hexa
				S	E(8b)	M(23b)	
19,5	10011,1	$1,00111 \times 2^4$	131	0	10000011	00111 <u>0...0</u> 18 fois	419C0000
-7,5							
-46,25							
0,3125							
							BE400000
							7F800000
0							
$-26,375 \times 2^{40}$							

PS : Si vous voulez vous entraîner chez vous, utilisez le programme C suivant (sans explications) :

```
#include <stdint.h>
#include <stdio.h>
main () { float f; union {uint32_t integer;float ieeeseptcinquatre;} n;
  printf("Entrez un nombre en base 10: ");scanf("%f",&f);
  n.ieeseptcinquatre=f;printf("IEEE754: 0x%08X\n",n.integer); }
```

9 Codage de texte

Objectif : *Comprendre le codage du texte*

1. Soit le texte (guillemets non-compris) « Les sanglots longs des violons de l'automne, blessent mon cœur d'une langueur monotone. » Est-il possible de représenter ce texte dans le jeu de caractères ASCII ?
2. Dans le jeu de caractère ISO-8859-15 (dit *latin-9*), il est possible de coder ce texte. Chaque caractère est alors codé par un octet unique. Quelle est la taille du fichier qui contient uniquement ce texte ?
3. Un polonais lit sur son vieil ordinateur le texte précédent. Il voit qu'une des lettres a été remplacée par " (c'est un double accent aigu, comme dans Erdős, et pas un tréma comme dans Gwenaël). Laquelle et pourquoi ? S'il renvoie le texte tel quel a son correspondant français du début, que verra le français et pourquoi ?

10 Le format UTF-8

Objectif : *Comprendre le codage du texte, comprendre un format, comprendre la différence entre jeu de caractères et codage des caractères.*

Le format UTF-8 est un codage des nombres entre 0x0000 et 0x1FFFFFF (prévu pour coder le jeu de caractère Unicode). C'est le codage suivant (les lettres en italique représentent des bits, qu'il faut reporter dans la troisième colonne) :

Valeurs	Écriture binaire	Codage UTF-8 (binaire)	taille UTF-8
0x0–0x7F	<i>abc defg</i>	0 <i>abc defg</i>	1 octet
0x80–0x7FF	<i>abc defg hijk</i>	110 <i>a bcde</i> 10 <i>fg hijk</i>	2 octets
0x800–0xFFFF	<i>abcd efgh ijkl mnop</i>	1110 <i>abcd</i> 10 <i>ef ghij</i> 10 <i>kl mnop</i>	3 octets
0x10000–0x1FFFFFF	<i>abcde fghi jklm nopq rstu</i>	11110 <i>abc</i> 10 <i>de fghi</i> 10 <i>jk lmno</i> 10 <i>pq rstu</i>	4 octets

Par exemple, le numéro 0x5D0 est encodée par la deuxième ligne (à cause de sa valeur). Traduit en binaire, il vaut 0b101 1101 0000. Il est donc codé sur deux octets : Le premier vaut 11010111, et le deuxième vaut 10010000. C'est-à-dire qu'il est codé par les deux octets 0xD7 et 0x90 (on écrit très souvent les octets en hexadécimal pour gagner de la place et facilité de lecture).

Vous veillerez à systématiquement écrire les résultats binaires aussi en hexadécimal !

1. Qu'est-ce qu'un jeu de caractère ? Que représente-t-on avec ?
2. Le caractère de numéro 0x0041 (A) est codé par quel(s) octet(s) en UTF-8 ?
3. Le caractère de numéro 0x00E9 (é) est codé par quel(s) octet(s) en UTF-8 ?
4. Le caractère de numéro 0x0F03 (Ꜧ) est codé par quel(s) octet(s) en UTF-8 ?
5. Le caractère de numéro 0x12084 (𐄀) est codé par quel(s) octet(s) en UTF-8 ?
6. Dans un fichier codé en UTF-8, on trouve les six octets suivants. Combien de caractères sont réellement codés dans ce texte ?

0xE6	0x9D	0x8c	0xDE	0xBC	0x43
------	------	------	------	------	------
7. L'anglais n'utilise que des caractères dont le numéro est dans la première ligne, et est codé traditionnellement en ISO-8859-1 (1 caractère = 1 octet). Le français utilise 5% de caractères de la deuxième ligne (le reste de la première), et est codé pareil (1 caractère = 1 octet). L'arabe (le russe, l'hébreu, le grec) sont aussi codés traditionnellement par 1 caractère = 1 octet, et comportent 95% de caractères de la deuxième ligne (le reste de la première ligne). Le chinois, en revanche est traditionnellement codé en BIG5 (1 caractère = 2 octets). Les textes chinois sont à 99% des caractères de la troisième ligne (le reste de la première ligne).
Depuis quelques années, les codages changent : on utilise de moins en moins le codage traditionnel et de plus en plus le codage UTF-8.
8. Pour un texte de 1000 caractères codé en UTF-8, combien d'octets seront utilisés en moyenne pour un texte anglais, français, russe et chinois ?
9. Quel est en chinois le pourcentage d'augmentation de la taille du texte par rapport au codage traditionnel ?

11 Comprendre un format

Objectif : *Maîtriser les conversions hexadécimal/binaire, comprendre un format complexe d'après sa description, connaître un format simple d'image noir et blanc*

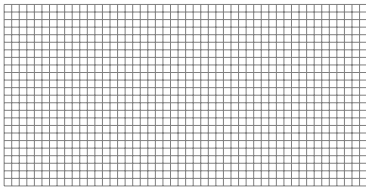
Cet exercice est l'étude du codage PBM qui code des images en noir et blanc. Une image est décrite comme un rectangle largeur × hauteur de carrés qui peuvent être soit noirs, soit blanc.

Le format est le suivant : D'abord le caractère P (codé comme dans la table ASCII donnée en cours) sur un octet, suivi du caractère 4. Ensuite, on trouve une série de caractères blancs (un ou plusieurs, voir ci-dessous pour la définition d'un caractère blanc). Ensuite vient la largeur de l'image, sous forme décimale, écrite en ASCII (un octet par chiffre), puis un ou plusieurs caractères blancs. Ensuite vient la hauteur de l'image (comme pour la largeur, en décimal et en ASCII) suivi d'un unique caractère blanc. La partie décrite jusque là s'appelle *l'entête* du format.

Après l'entête, on va coder bit par bit les pixels de l'image. Un pixel blanc se code par un 0, un noir par un 1. Si la largeur de la ligne n'est pas divisible par 8, on fait comme si l'image était plus large que réellement (on va jusqu'au multiple de 8 immédiatement supérieur, et tout ce qui ne fait pas partie de l'image est considéré blanc), mais bien sûr on ne modifie pas dans l'entête la largeur de l'image (on la laisse à sa valeur réelle).

Les caractères blancs sont les caractères de code hexadécimal 0x09, 0x0A, 0x0D ou 0x20. Ils correspondent au caractère de tabulation (TAB), au caractère de nouvelle ligne (LF), au caractère de retour chariot (CR) et à l'espace (SP). Votre enseignant pourra vous expliquer l'utilisation habituelle de ces caractères.

1. En utilisant votre cours, trouvez le code du caractère P et le code du caractère 4. Donnez-le en hexadécimal, en binaire, en octal, en décimal.
2. Combien de bits sont nécessaires pour coder de cette façon une image de taille 8 pixels de large et 1 de haut ? Et pour une image de 1024 pixels de large et 1024 de haut ? Donnez à chaque fois la taille de l'entête et la taille totale.



3. Le contenu d'un fichier (représenté ici en hexadécimal) est comme suit :

ADRESSE	OCTETS EN HEXADÉCIMAL
00000000	50 34 0a 31 36 20 31 36 0a 00 04 06 ca 09 55 11
00000010	23 10 04 20 04 40 04 40 04 46 08 49 30 28 a0 68
00000020	a0 38 e0 34 e0 3f fc 07 ee

Transformez chacun des cinq premiers octets en utilisant la table ASCII du cours pour faire le début de déchiffrement du fichier. Est-ce qu'il y a des octets de ce fichier qu'on ne peut pas transformer ainsi ?

4. Quelle est la dimension de cette image ?
5. Quelle est la taille de ce fichier ? Comment se répartit entête et corps ?
6. En utilisant le quadrillage ci-dessus, décidez le fichier. Que représente cette image ?