Représentation des données – Arithmétique entière

Tanmoy MONDAL

tanmoy.mondal@lirmm.fr
Diapos de D. Delahaye et Chouki TIBERMACINE

Objectifs du cours

1. Comprendre le principe des codages binaires ;
2. Savoir représenter des nombres entiers dans diverses représentations ;
3. Savoir représenter des nombres réels (prochain cours) ;
4. Savoir calculer avec les représentations des nombres.

Différents types d'informations

- Instructions ;
- Données :
  - Nombres (entiers, réels) ;
  - Caractères et chaînes ;
  - Images ;
  - Sons et vidéos.

Toujours représentées sous forme binaire (0 ou 1) à l'aide de bits.

Codage

Fonction établissant une correspondance entre la représentation externe de l'information (par exemple, 'A', 36, un son, ou une image) et sa représentation interne qui est une suite de bits (par exemple, 01000001, 100100, etc.).

Système binaire

Un système de numération utilisant la base 2. Toutes les informations sont codées avec des 0 et des 1.

- 1 bit : 2^1 possibilités = 0, 1 ;
- 2 bits : 2^2 possibilités = 00, 01, 10, 11 ;
- n bits : 2^n possibilités ;

Système binaire -suite-

- Un mot de taille n = un ensemble de bit avec un poids $2^{n-1} \ldots 2^1, 2^0$;

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	1	1	0	0

Avantages

- Facile à réaliser techniquement. En électronique, ces 2 états correspondent à l'existence ou non d'une tension (+5V=1 et 0V=0).
- Opérations fondamentales faciles à effectuer (circuits logiques).
- Arithmétique binaire réalisée à partir de la logique symbolique.

Système binaire -suite-

- Un mot de taille n = un ensemble de bit avec un poids $2^{n-1} \ldots 2^1, 2^0$;

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	1	1	0	0

Exercice

L'opération puissance permet de connaître le nombre de mots possibles pour un nombre *n* de bits. Mais quelle opération permet de faire l'inverse, c'est-à-dire de savoir combien il faut de bits pour représenter *n* possibilités ?
Par exemple, combien de bits faudra-t-il pour représenter 256 possibilités ?

Système binaire -suite-

- Un mot de taille n = un ensemble de bit avec un poids $2^{n-1} \ldots 2^1, 2^0$;

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	1	1	0	0

Solution

- n bits $\Rightarrow 2^n$ possibilités (donc 2^n mots) ;
- x possibilités $\Rightarrow log_2(x) = log_2(2^y) = y$ bits ;
- Si x n'est pas une puissance de 2, prendre le plus petit y t.q. $2^y > x$.

Pour 256 possibilités, il faut donc 8 bits : $log_2(256) = log_2(2^8) = 8$.

Représentation de l'information

- Instructions :
  - Le codage dépend du processeur ;
  - Décodage par l'unité de commande.
  - Nombre limité d'instructions (processeur CISC/RISC).
- Données :
  - Non numériques (codage assez simple car aucune opération arithmétique ou logique ne sera appliquée sur ces données, une table de correspondance suffit) ;
  - Numériques (codage complexe qui doit faciliter la mise en place de circuits réalisant les opérations arithmétiques).

Données non numériques

Afin de faciliter les échanges entre machines, des codages binaires normalisés ont été établis (BCD, ASCII, Unicode, etc.).

- Nombre variable de bits, 6, 7, 8, 16, 32 ;
- Certains bits réservés au contrôle des autres.

Certains codages ne sont pas normalisés. C'est pourquoi l'on peut rencontrer des problèmes lors d'échanges de fichiers comportant des caractères spéciaux/accentués (entre Mac et PC par exemple).

Caractères ASCII

- ASCII = American Standard Code for Information Interchange ;
- 7 bits (⇒ 128 caractères) ;
- Pas de caractères accentués (accents : ISO 8859-1, 8 bits) ;
- 1 bit supplémentaire utilisé pour le contrôle de parité (si voulu).

0 : nul	1 : soh	2 : stx	3 : etx	4 : eot	5 : enq	6 : ack	7 : bel
8 : bs	9 : ht	10 : nl	11 : vt	12 : np	13 : cr	14 : so	15 : si
16 : dle	17 : dc1	18 : dc2	19 : dc3	20 : dc4	21 : nak	22 : syn	23 : etb
24 : can	25 : em	26 : sub	27 : esc	28 : fs	29 : gs	30 : rs	31 : us
32 : sp	33 : !	34 : "	35 : #	36 : $	37 : %	38 : &	39 : '
40 : (	41 :)	42 : *	43 : +	44 : ,	45 : -	46 : .	47 : /
48 : 0	49 : 1	50 : 2	51 : 3	52 : 4	53 : 5	54 : 6	55 : 7
56 : 8	57 : 9	58 : :	59 : ;	60 : <	61 : =	62 : >	63 : ?
64 : @	65 : A	66 : B	67 : C	68 : D	69 : E	70 : F	71 : G
72 : H	73 : I	74 : J	75 : K	76 : L	77 : M	78 : N	79 : O
80 : P	81 : Q	82 : R	83 : S	84 : T	85 : U	86 : V	87 : W
88 : X	89 : Y	90 : Z	91 : [	92 :	93 :]	94 :	95 : _
96 : '	97 : a	98 : b	99 : c	100 : d	101 : e	102 : f	103 : g
104 : h	105 : i	106 : j	107 : k	108 : l	109 : m	110 : n	111 : o
112 : p	113 : q	114 : r	115 : s	116 : t	117 : u	118 : v	119 : w
120 : x	121 : y	122 : z	123 : {	124 : |	125 : }	126 :	127 : del

7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	**0**0000000	**1**0000000
1010001	3	**1**1010001	**0**1010001
1101001	4	**0**1101001	**1**1101001
1111111	7	**1**1111111	**0**1111111

Caractères Unicode

- 1 à 4 octets (⇒ 1 114 112 caractères) ;
- Codage unique quelle que soit la plate-forme, le logiciel, la langue ;
- Inclus des propriétés fonctionnelles et des renseignements linguistiques ou typographiques ;
- Normalisé ISO/IEC 10646 (UTF-8, UTF-16, UTF-32) ;

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A		„ 201E	… 2026	† 2020	‡ 2021		‰ 2030		‹ 2039		¨ 00A8	ˇ 02C7	¸ 00B8
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122		› 203A		¯ 00AF	˛ 02DB	
A0	NBSP 00A0		¢ 00A2	£ 00A3	¤ 00A4		¦ 00A6	§ 00A7	Ø 00D8	© 00A9	Ŗ 0156	« 00AB	¬ 00AC	- 00AD	® 00AE	Æ 00C6
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	ø 00F8	¹ 00B9	ŗ 0157	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	æ 00E6
C0	Ą 0104	Į 012E	Ā 0100	Ć 0106	Ä 00C4	Å 00C5	Ę 0118	Ē 0112	Č 010C	É 00C9	Ź 0179	Ė 0116	Ģ 0122	Ķ 0136	Ī 012A	Ļ 013B
D0	Š 0160	Ń 0143	Ņ 0145	Ó 00D3	Ō 014C	Õ 00D5	Ö 00D6	× 00D7	Ų 0172	Ł 0141	Ś 015A	Ū 016A	Ü 00DC	Ż 017B	Ž 017D	ß 00DF
E0	ą 0105	į 012F	ā 0101	ć 0107	ä 00E4	å 00E5	ę 0119	ē 0113	č 010D	é 00E9	ź 017A	ė 0117	ģ 0123	ķ 0137	ī 012B	ļ 013C
F0	š 0161	ń 0144	ņ 0146	ó 00F3	ō 014D	õ 00F5	ö 00F6	÷ 00F7	ų 0173	ł 0142	ś 015B	ū 016B	ü 00FC	ż 017C	ž 017E	˙ 02D9

Données numériques

- Nombres entiers non signés (positifs ou nuls) : 0 ; 1; 315 ;
- Nombres entiers signés (positifs et négatifs) : -1; -1255 ;
- Nombres flottants (réels) : 3,1415 ; -0,5 (prochain cours).

Un algorithme de codage réalise la conversion en binaire.
Les opérations arithmétiques (+, -, *, /) sont effectuées exclusivement en arithmétique binaire.

Entiers non signés

Représentation en base b

Tout nombre entier positif (de n chiffres a_i) peut être représenté, en base b, par une expression de la forme :

$$x = a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} + \ldots + a_1 * b^1 + a_0 * b^0$$

7843 en base 10

$7843 = 7 * 10^3 + 8 * 10^2 + 4 * 10^1 + 4 * 10^1$

$$\Rightarrow 7843$$

335 en base 2

$335 = 1 * 2^8 + 0 * 2^7 + 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0$

$$\Rightarrow 101001111$$

Système de numération octal

- Utilise huit chiffres, 0,1,2,3,4,5,6,7.
- Aussi appelé système de base 8

Step	Octal Number	Decimal Number
Step 1	12570_8	$((1 \times 8^4) + (2 \times 8^3) + (5 \times 8^2) + (7 \times 8^1) + (0 \times 8^0))_{10}$
Step 2	12570_8	$(4096 + 1024 + 320 + 56 + 0)_{10}$
Step 3	12570_8	5496_{10}

Système de numération Hexadécimal

- Utilise 10 chiffres et 6 lettres, 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F.
- Les lettres représentent des nombres à partir de 10. A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.
- Aussi appelé système de base 16.

Step	Hexadecimal Number	Decimal Number
Step 1	$19FDE_{16}$	$((1 \times 16^4) + (9 \times 16^3) + (F \times 16^2) + (D \times 16^1) + (E \times 16^0))_{10}$
Step 2	$19FDE_{16}$	$((1 \times 16^4) + (9 \times 16^3) + (15 \times 16^2) + (13 \times 16^1) + (14 \times 16^0))_{10}$
Step 3	$19FDE_{16}$	$(65536 + 36864 + 3840 + 208 + 14)_{10}$
Step 4	$19FDE_{16}$	106462_{10}

Changement de base : binaire $\longrightarrow$ décimal (décodage)

Méthode

Additionner les puissances de 2 correspondants aux bits de valeur 1.

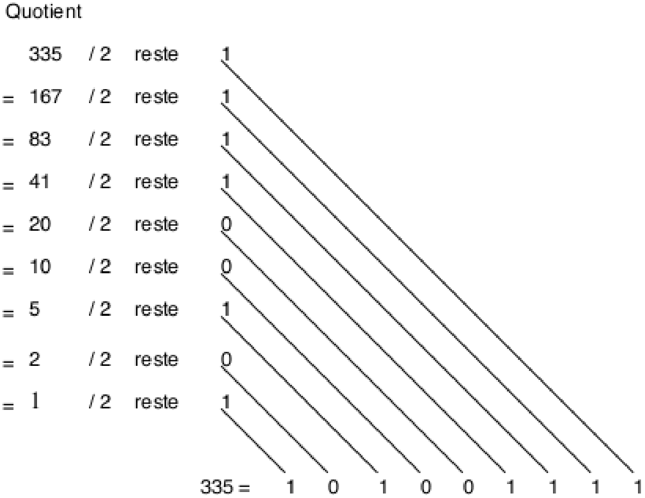
Exemple

$101001111 = 2^8 + 2^6 + 2^3 + 2^2 + 2^1 + 2^0 = 256 + 64 + 8 + 4 + 2 + 1 = 335$

Attention au poids !

- Bit de poids faible : bit ayant la moindre valeur (celui de droite) ;
- Bit de poids fort : bit ayant la plus grande valeur (celui de gauche).

Changement de base : décimal → binaire (encodage)



Exercices

Décodage

Combien valent en décimal les nombres binaires suivants :

1. 1010000100 ;
2. 0100011111.

Encodage

Combien valent en binaire les nombres décimaux suivants :

1. 365 ;
2. 171.

Addition de nombres binaires

222 + 17 en base 2

$$
\begin{array}{cccccccccc}
 & & 1 & 1 & {}^{1}0 & 1 & 1 & 1 & 1 & 0 \\
+ & & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
 & & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1
\end{array}
$$

Case	A + B	Sum	Carry
1	0 + 0	0	0
2	0 + 1	1	0
3	1 + 0	1	0
4	1 + 1	0	1

$\Rightarrow$ 239.

222 + 199 en base 2

$$
\begin{array}{cccccccccc}
 & {}^{1} & {}^{1}1 & 1 & {}^{1}0 & {}^{1}1 & {}^{1}1 & {}^{1}1 & 1 & 0 \\
+ & & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}
$$

Case	A + B	Sum	Carry
1	0 + 0	0	0
2	0 + 1	1	0
3	1 + 0	1	0
4	1 + 1	0	1

$\Rightarrow$ 421 ;

$\Rightarrow$ Attention au bit de retenue (« carry »).

Soustraction de nombres binaires

8 — 1 en base 2

$$
\begin{array}{ccccc}
 & 1 & {}_1 0 & {}_1 0 & {}_1 0 \\
- & 0 & 0 & 0 & 1 \\
 & 1 & 1 & 1 & \\
\hline
 & 0 & 1 & 1 & 1
\end{array}
$$

Case	A	-	B	Subtract	Borrow
1	0	-	0	0	0
2	1	-	0	1	0
3	1	-	1	0	0
4	0	-	1	0	1

$\Rightarrow$ 7.

13 — 6 en base 2

$$
\begin{array}{ccccc}
 & 1 & {}_1 1 & {}_1 0 & 1 \\
- & 0 & 1 & 1 & 0 \\
 & 1 & 1 & & \\
\hline
 & 0 & 1 & 1 & 1
\end{array}
$$

Case	A	-	B	Subtract	Borrow
1	0	-	0	0	0
2	1	-	0	1	0
3	1	-	1	0	0
4	0	-	1	0	1

$\Rightarrow$ 7.

Multiplication et division de nombres binaires

7 × 5 en base 2

$$
\begin{array}{rcccccc}
 & & & & 1 & 1 & 1 \\
\times & & & & 1 & 0 & 1 \\
\hline
 & & & & 1 & 1 & 1 \\
 & {}^{1} & {}^{1} & {}^{1}0 & 0 & 0 & 0 \\
+ & & 1 & 1 & 1 & 0 & 0 \\
\hline
 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

$\Rightarrow$ 35.

356/4 en base 2

$$
\begin{array}{rccccccccc}
 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
/ & & & & & & & 1 & 0 & 0 \\
\hline
 & & & 1 & 0 & 1 & 1 & 0 & 0 & 1
\end{array}
$$

$\Rightarrow$ 89.

Multiplication et division de nombres binaires

7 × 5 en base 2

$$
\begin{array}{ccccccc}
 & & & & 1 & 1 & 1 \\
\times & & & & 1 & 0 & 1 \\
\hline
 & & & & 1 & 1 & 1 \\
 & {}^{1} & {}^{1} & {}^{1}0 & 0 & 0 & 0 \\
+ & & 1 & 1 & 1 & 0 & 0 \\
\hline
 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

$\Rightarrow$ 35.

Exercice

Dérouler la division de 356 par 4.

Taille fixe

L'entier maximal pouvant être codé dépendra du nombre de bits que l'on réserve pour coder un nombre. En général, les entiers sont codés sur un mot. Par exemple, pour un ordinateur 32 bits :
$2^{32} - 1 = 4\,294\,967\,295$.

Dépassement de capacité (« Overflow »)

Se produit lorsque par exemple le résultat d'une opération sur des nombres produit un nombre plus grand que la taille du mot prévu pour représenter ces nombres (bit de retenue).

Nombres binaires non signés

Principe

- Les bits présents dans le nombre binaire non signé contiennent la magnitude d'un nombre.
- Si le nombre binaire non signé contient "N" bits, tous les N bits représentent la magnitude du nombre, car il n'a pas de bit de signe

$108_{10} = 1101100_2$

Ces 7 bits représentent la magnitude du nombre 108.

Entiers signés

Principe

- Taille de la représentation connue ;
- Nombres positifs codés comme précédemment ;
- Si le nombre binaire signé contient 'N' bits, alors N-1 bits ne représentent que la magnitude du nombre
- Bit n^{eme} pour le signe (signe + = 0 , signe - = 1) ; ("bit le plus significatif")
- Bits $\{n - 1...1\}^{eme}$ pour les positifs et leurs compléments.
- Négatifs sont représentés par le complément à 2 ;
  - Forme de magnitude de signe
  - 1 complément
  - 2 complément

Entiers signés

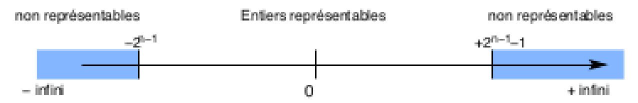
Complément à 2

- Complément à 1 (inverser les bits) : $6 = 0110 \longrightarrow 1001$;
- Complément à 2 (complément à 1 + 1) :
  $6 = 0110 \longrightarrow 1001 + 1 \longrightarrow 1010 = -6$;
- Méthode rapide : localisez le premier 1 à droite, laissez-le inchangé, puis prenez le complément du reste des bits qui est à gauche : $0110 \longrightarrow 1010$;

Entiers signés

Nombres représentables

- Sur 4 bits : $6 = 0110$, $-6 = 1001 + 1 = 1010$;
- Sur 4 bits : $0 = 0000$ (0 est positif dans cette représentation).



pour n=8, [-128 ; 127]

Exercice

Comment représenter -8 en complément à 2 sur 4 bits ?

Entiers signés

Exercice

Quelle est la représentation binaire sur 8 bits des décimaux suivants :

1. 112 ;
2. -112.

Quelle est la valeur décimale des entiers binaires sur 8 bits suivants :

1. 1111 1111 ;
2. 1000 0000.

Opérations arith. avec les nombres en comp. à 2

Addition

4 cas sont possibles... :

- Les deux nombres sont positifs :
  - Addition binaire classique et on oublie la dernière retenue (à gauche).
  - Le bit de signe à 0 indique la somme résultante est positifs
  - L'addition de deux nombres positifs donnera un autre nombre positifs.

Opérations arith. avec les nombres en comp. à 2

Addition

4 cas sont possibles... :

- Les deux nombres sont négatifs :
  - Nous prenons le complément de 2 de chaque nombre
  - Addition binaire classique et on oublie la dernière retenue (à gauche).
  - Le bit de signe à 1 indique la somme résultante est négative
  - L'addition de deux nombres négatifs donnera un autre nombre négatif.
  - En prenant le complément de 2, nous aurons l'magnitude de la somme résultante (**n'oublie pas**)

Opérations arith. avec les nombres en comp. à 2

Addition

4 cas sont possibles... :

- Le nombre positif est plus grand (ou égal) que le nombre négatif :
  - Nous prenons le complément de 2 de nombre négatif
  - Addition binaire classique et on oublie la dernière retenue (à gauche).
  - Le bit de signe à 0 indique la somme résultante est positifs

Opérations arith. avec les nombres en comp. à 2

Addition

4 cas sont possibles... :

- Le nombre négatif est plus grand que le nombre positif : :
  - Nous prenons le complément de 2 de nombre négatif
  - Addition binaire classique et on oublie la dernière retenue (à gauche).
  - Le bit de signe à 1 indique la somme résultante est négatif

Opérations arith. avec les nombres en comp. à 2

Soustraction

La soustraction est considérée comme un cas particulier de l'addition :

- A - B = A + (-B) ;
- -A - B = (-A) + (-B).

On prend donc le système complément à deux pour représenter les négatifs, et on effectue une addition (voir transparent précédent).

Opérations arith. avec les nombres en comp. à 2

Multiplication

Les deux nombres doivent être représentés dans une forme sans complément (en valeur absolue). On effectue la multiplication et on décide du signe du résultat :

- Opérandes de même signe : le résultat est positif ;
- Opérandes de signes différents : le résultat est négatif, on le représente avec son complément à 2 ;
- Il peut y avoir dépassement de capacité dans les deux cas.

Opérations arith. avec les nombres en comp. à 2

Division

Les deux nombres doivent être représentés dans une forme sans complément (en valeur absolue) :

1. Déterminer si le dividende et le diviseur sont de mêmes signes ou de signes différents. Ceci va déterminer le signe du quotient ; initialiser le quotient à 0.
2. Soustraire le diviseur du dividende en utilisant l'addition avec complément à deux pour obtenir le premier reste partiel ; incrémenter le quotient. Si le reste partiel est positif aller à l'étape trois. Si le reste partiel est 0 ou négatif la division est terminée.
3. Soustraire le diviseur du reste partiel et incrémenter le quotient. Si le résultat est positif, répéter l'opération pour le reste partiel trouvé. Si le résultat est 0 ou négatif la division est terminée.

Opérations arith. avec les nombres en comp. à 2

Exercice

Additionner en binaire sur 8 bits les nombres décimaux suivants :

1. 127 et 1 ;
2. 127 et -1.

Soustraire en binaire sur 8 bits les nombres décimaux suivants :

1. -1 et 1 ;
2. -1 et -128.

Remerciements et références

Remerciements

David Delahaye, professeur à la FDS (mon prédécesseur)
Chouki TIBERMACINE, MCF à PolyTech-Montpellier (mon prédécesseur)

Références bibliographiques

- Paolo Zanella, Yves Ligier et Emmanuel Lazard. Architecture et technologie des ordinateurs - 6e éd. - Cours et exercices corrigés. Septembre 2018