

## Objectifs

- Comprendre les codages binaires, décimaux, et hexadécimaux
- Appréhender le codage des caractères (ASCII et ASCII étendu)

## 1 Codage décimal (base 10)

- Que signifie " base 10 " ?
- La numération romaine est-elle en base 10 ?

## 2 Codage binaire (base 2)

Ici chaque élément peut coder 2 valeurs (0 ou 1) là où le système décimal encode 10 valeurs différentes dans chaque élément (0 à 9). Pour la justification de l'utilisation du binaire, allez voir la partie *Why Binary?* dans la partie Approfondissement en fin de TD.

- Que signifie numération binaire positionnelle ?

### 2.1 Du binaire au décimal

- Partez de cette définition pour compléter le tableau ci-dessous et convertir en base décimale les nombres exprimés en base 2 :

Binaire	Analyse	Décimal
0		
101	$1 * 2^2 + 0 * 2^1 + 1 * 2^0$	5
1		
110010		
111101		

### 2.2 Du décimal au binaire

- En utilisant la méthodologie proposée ci-dessous, convertissez en binaire la série de nombres en base 10 qui suit :

**Méthodologie :**

Binaire	0	0	0	0	0	<b>1</b>	0	0	<b>1</b>
Décomposition	$1 * 2^9$	$1 * 2^8$	$1 * 2^7$	$1 * 2^6$	$1 * 2^5$	$1 * 2^4$	$1 * 2^3$	$1 * 2^1$	$1 * 2^0$
Valeur décimale	256	128	64	32	16	<b>8</b>	4	2	<b>1</b>

*Lecture : Si je veux représenter 9 je dois trouver la somme correspondante dans la ligne des valeurs décimales. En l'occurrence, c'est  $8 + 1$ . Je peux donc remonter à la ligne Binaire pour récupérer les "1" dont j'ai besoin, le reste ce sont des "0" :  $8_{10} = 000001001_{12}$*

**Tableau à compléter :**

Décimal	Analyse	Binaire
0		
1		
2		
3		
4		
5		
6		
7		
8		
9	$9 = 8 + 1 = 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0$	1001
32		
66		
128		
255		

- Quel est l'entier maximal que l'on puisse coder sur 16 bits ?

## 3 Addition en Binaire

### 3.1 Additions en binaire

Règles d'additions en binaire:

- $0 + 0 = 0$ , on pose 0
- $0 + 1 = 1$ , on pose 1
- $1 + 1 = 10$  : on pose 0 et on retient 1
- $1 + 1 + retenue = 11$  : on pose 1 et on retient 1

Explication pour  $1010_{(2)} + 0011_{(2)}$  (on lit de droite à gauche):

	1	0	1	0
+	0	0	1	1
Opérations	on pose <b>1</b>	0 + la <b>retenue</b> , on pose <b>1</b>	on pose <b>0</b> et on <b>retient</b> 1	on pose <b>1</b>
Résultat	1	1	0	1

On a bien le résultat correspondant en système décimal :  $10_{(10)} + 3_{(10)} = 13_{(10)}$

- A vous pour :  $1111_{(2)} + 1111_{(2)}$

### 3.2 Conversion et addition

- Convertissez ces nombres en binaire, et additionnez les :  $19_{(10)} + 71_{(10)}$

Table 1: Table d'addition hexadécimale

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

## 4 Codage hexadécimal (base 16)

- Quels sont les 16 chiffres du système hexadécimal ?
- Convertissez  $4D5_{(16)}$ , et  $F48A_{(16)}$  en décimal.
- Addition en hexadécimal. En vous appuyant sur le tableau ??, combien vaut  $D + C$  en base 16 ? Exprimez ensuite ce calcul en base 10.

## 5 Codage des caractères (Codage ASCII)

- Combien de codes différents existe-t-il dans l'ASCII standard (cf. Tableau ?? page ???)
- Ce qui signifie qu'ils sont codés sur combien de bits ?
- Observez la table ci-après. Quel problème peut survenir lorsqu'on écrit en français?
- Quelle solution a été adoptée par l'ISO pour résoudre ce problème ? Quelle conséquence ?

Figure 1: Table ASCII (source : <http://www.asciitable.com/>)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	##32;	<b>Space</b>	64	40	100	##64;	<b>@</b>	96	60	140	##96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	##33;	<b>!</b>	65	41	101	##65;	<b>A</b>	97	61	141	##97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	##34;	<b>"</b>	66	42	102	##66;	<b>B</b>	98	62	142	##98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	##35;	<b>#</b>	67	43	103	##67;	<b>C</b>	99	63	143	##99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	##36;	<b>\$</b>	68	44	104	##68;	<b>D</b>	100	64	144	##100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	##37;	<b>%</b>	69	45	105	##69;	<b>E</b>	101	65	145	##101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	##38;	<b>&amp;</b>	70	46	106	##70;	<b>F</b>	102	66	146	##102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	##39;	<b>'</b>	71	47	107	##71;	<b>G</b>	103	67	147	##103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	##40;	<b>(</b>	72	48	110	##72;	<b>H</b>	104	68	150	##104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	##41;	<b>)</b>	73	49	111	##73;	<b>I</b>	105	69	151	##105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	##42;	<b>*</b>	74	4A	112	##74;	<b>J</b>	106	6A	152	##106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	##43;	<b>+</b>	75	4B	113	##75;	<b>K</b>	107	6B	153	##107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	##44;	<b>,</b>	76	4C	114	##76;	<b>L</b>	108	6C	154	##108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	##45;	<b>-</b>	77	4D	115	##77;	<b>M</b>	109	6D	155	##109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	##46;	<b>.</b>	78	4E	116	##78;	<b>N</b>	110	6E	156	##110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	##47;	<b>/</b>	79	4F	117	##79;	<b>O</b>	111	6F	157	##111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	##48;	<b>0</b>	80	50	120	##80;	<b>P</b>	112	70	160	##112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	##49;	<b>1</b>	81	51	121	##81;	<b>Q</b>	113	71	161	##113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	##50;	<b>2</b>	82	52	122	##82;	<b>R</b>	114	72	162	##114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	##51;	<b>3</b>	83	53	123	##83;	<b>S</b>	115	73	163	##115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	##52;	<b>4</b>	84	54	124	##84;	<b>T</b>	116	74	164	##116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	##53;	<b>5</b>	85	55	125	##85;	<b>U</b>	117	75	165	##117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	##54;	<b>6</b>	86	56	126	##86;	<b>V</b>	118	76	166	##118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	##55;	<b>7</b>	87	57	127	##87;	<b>W</b>	119	77	167	##119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	##56;	<b>8</b>	88	58	130	##88;	<b>X</b>	120	78	170	##120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	##57;	<b>9</b>	89	59	131	##89;	<b>Y</b>	121	79	171	##121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	##58;	<b>:</b>	90	5A	132	##90;	<b>Z</b>	122	7A	172	##122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	##59;	<b>;</b>	91	5B	133	##91;	<b>[</b>	123	7B	173	##123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	##60;	<b>&lt;</b>	92	5C	134	##92;	<b>\</b>	124	7C	174	##124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	##61;	<b>=</b>	93	5D	135	##93;	<b>]</b>	125	7D	175	##125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	##62;	<b>&gt;</b>	94	5E	136	##94;	<b>^</b>	126	7E	176	##126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	##63;	<b>?</b>	95	5F	137	##95;	<b>_</b>	127	7F	177	##127;	<b>DEL</b>

Source: [www.LookupTables.com](http://www.LookupTables.com)

## Approfondissement

- Pourquoi choisir le binaire pour les ordinateurs (*Why Binary ?*) ? <https://www.youtube.com/watch?v=thrx3SBEPtL8>
- Un convertisseur Binaire-Décimal-Hexadécimal :  
<http://sebastienguillon.com/test/javascript/convertisseur.html>
- Un exemple de problème d'encodage (ou *encoding*) :

	Encoding:Utf-8	Encoding:ISO-8859-1 (latin1)
Interprété comme:Utf-8	Étonnant que ça ait été si délicat à faire.	Étonnant que a ait t si délicat faire.
Interprété comme:Latin-1	Ã<89>tonnant que Ãa ait Ã©tÃ© si dÃ©licat Ã faire.	Étonnant que ça ait été si délicat à faire.