

Codage

E. Jeandel

Emmanuel.Jeandel at lif.univ-mrs.fr

Représentation des données

- Comment coder une image en un fichier ?
- Comment coder un texte en un fichier ?
- Comment représenter une couleur dans un ordinateur ?
- Comment représenter un graphe dans un ordinateur ?
- Comment représenter une base de données dans un ordinateur ?

- La notion de base est le *bit*.
- Un bit peut prendre deux valeurs, 0 ou 1.
- Les bits sont regroupés, pour simplifier, par 8, pour former ce qu'on appelle un *octet*.

Représenter des données, c'est donc les représenter comme une série de bits, ou comme une série d'octets.

Représentation des nombres

Un nombre entier est représenté par son écriture en base 2 :

51		110011
51		00110011
1664		11010000000

Plus d'informations dans le cours d'Architecture en L3

Représentation des dates

Plusieurs formats :

- Format utilisé (entre autres) sous DOS et Windows :
 - Date sur 16 bits : 5 pour le jour, 4 pour le mois, et 7 pour l'année, en prenant comme référence 1980

$$2010/01/19 = 0011110 \cdot 0001 \cdot 10011 = 0x3c33$$

Bug de l'an... ?

- Heure sur 16 bits : 5 pour l'heure, 6 pour les minutes, 5 pour les secondes. **Problème ?**

$$9 : 51 : 36 = 01001 \cdot 110011 \cdot 10010 = 0x4e72$$

- Format utilisé sous Unix : nombre de secondes écoulées depuis minuit UTC (temps universel coordonnée) le 1er janvier 1970, codé sur 32 bits, dont un bit pour le signe.

$$2010/01/19 \text{ à } 9 :51 :36 = 1263891096$$

Bug de l'an... ? (il y a $31557600 \sim 15 \times 2^{21}$ secondes dans un an)

Représentation des caractères

Un caractère est représenté par 8 bits (donc par un nombre entre 0 et 255). Des tables de correspondance expliquent comment on passe des 8 bits au caractère correspondant.

Exemple pour le caractère “é” :

Norme	Code binaire	Décimal
ISO/IEC 8859-1 (Latin-1 Western European)	11101001	233
Mac OS Roman	10001110	142
CP437	10000010	130

99% des normes ont les mêmes 128 premiers caractères, correspondant à la norme ASCII.

Chacune des normes ne permet de représenter que 256 caractères : insuffisant pour certaines langues.

Codepage 819 - Latin 1 - ISO 8859-1

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1-	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2-		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
7-	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	007F
8-																
	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	008A	008B	008C	008D	008E	008F
9-																
	0090	0091	0092	0093	0094	0095	0096	0097	0098	0099	009A	009B	009C	009D	009E	009F
A-	;	€	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	-	
	00A0	00A1	00A2	00A3	00A4	00A5	00A6	00A7	00A8	00A9	00AA	00AB	00AC	00AD	00AE	00AF
B-	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
	00B0	00B1	00B2	00B3	00B4	00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF
C-	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
	00C0	00C1	00C2	00C3	00C4	00C5	00C6	00C7	00C8	00C9	00CA	00CB	00CC	00CD	00CE	00CF
D-	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
	00D0	00D1	00D2	00D3	00D4	00D5	00D6	00D7	00D8	00D9	00DA	00DB	00DC	00DD	00DE	00DF
E-	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
	00E0	00E1	00E2	00E3	00E4	00E5	00E6	00E7	00E8	00E9	00EA	00EB	00EC	00ED	00EE	00EF
F-	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
	00F0	00F1	00F2	00F3	00F4	00F5	00F6	00F7	00F8	00F9	00FA	00FB	00FC	00FD	00FE	00FF

Codepage 437 - United States

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-																
	263A	263B	2665	2666	2663	2660	2022	25D8	25C8	25D9	2642	2640	266A	266B	263C	
1-																
	25BA	25C4	2195	203C	00B6	00A7	25AC	21A8	2191	2193	2192	2190	221F	2194	25B2	25BC
2-																
	0030	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
7-	p	q	r	s	t	u	v	w	x	y	z	{	 	}	~	
	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	2300
8-	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
	00C7	00FC	00E9	00E2	00E4	00E0	00E5	00E7	00EA	00EB	00E8	00EF	00EE	00EC	00C4	00C5
9-	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	€	£	¥	Ps	f
	00C9	00EB	00CB	00F4	00F6	00F2	00FB	00F9	00FF	00D6	00DC	00A2	00A3	00A5	20A7	0180
A-	á	í	ó	ú	ñ	Ñ	a	o	¿			½	¼	;	«	»
	00E1	00ED	00F3	00FA	00F1	00D1	00AA	00BA	00BF	2310	00AC	00BD	00BC	00A1	00AB	00BB
B-																
	2091	2092	2193	2502	2524	2561	2562	2556	2555	2563	2551	2557	255D	255C	255B	2510
C-																
	2514	2534	252C	251C	2500	253C	255E	255F	255A	2554	2509	2566	2560	2550	256C	2567
D-																
	2568	2564	2565	2559	2558	2552	2553	256B	256A	2516	250C	2568	2564	256C	2590	2560
E-	α	β	Γ	π	Σ	σ	μ	τ	Φ	Θ	Ω	δ	∞	φ	ε	
	03B1	00DF	0393	03C0	03A3	03C3	03B3	03C4	03A6	0398	03A9	03B4	221E	03C6	03B5	2229
F-													²		2	
	2261	00B1	2265	2264	2320	2321	00F7	2248	00B0	2219	00B7	221A	207F	00B2	25A0	00A0

Codepage 1275 - Apple Latin 1

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F	
0-		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F	
1-		⌘ 0010	√ 2318 2713	◆ 2666	IBM FBFF												
2-		!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
3-		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4-		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5-		P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6-		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7-		p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8-		Ä	Å	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã	å	ç	é	è
9-		ê	ë	í	ì	î	ï	ñ	ó	ò	ô	ö	õ	ú	ù	û	ü
A-		†	°	¢	£	§	•	¶	ß	®	©	™	'	”	≠	Æ	Ø
B-		∞	±	≤	≥	¥	μ	∂	Σ	Π	π	∫	∂	∂	Ω	æ	ø
C-		¿	¡	¬	√	f	≈	Δ	«	»	...	À	Ã	Ö	Œ	œ	
D-		-	—	“	”	‘	’	÷	◇	ÿ	ÿ	/	▣	<	>	fi	fl
E-		‡	·	,	”	‰	Â	Ê	Á	Ë	È	Í	Î	Ï	Ì	Ó	Ô
F-		IBM FBFF	Ò	Ú	Û	Ü	ı	ˆ	˜	˘	˙	˚	¸	˝	˞	˟	ˠ

Codepage 924 - Latin 9 - EBCDIC

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		0001	0002	0003	000C	0005	0006	0007	0007	000D	000E	000B	000C	000D	000E	000F
1-	0010	0011	0012	0013	000D	0005	0016	0007	0018	0019	0002	000F	001C	001D	001E	001F
2-	0080	0081	0082	0083	0084	000A	0017	0018	0088	0089	008A	008B	008C	0005	0006	0007
3-	0080	0091	0016	0093	0084	0095	0086	0004	0088	0099	008A	008B	0014	0015	009E	001A
4-	0020	00A0	â	ä	à	á	ã	ä	ç	ñ	Ý	.	<	(+	!
5-	0026	00E9	ê	ë	è	í	î	ï	ì	ß	l	\$	*)	;	^
6-	002D	002F	Â	Ä	À	Á	Ã	Ä	Ç	Ñ	Š	,	%	_	>	?
7-	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	`	:	#	@	'	=	"
8-	Ø	a	b	c	d	e	f	g	h	i	«	»	ð	ý	þ	±
9-	°	j	k	l	m	n	o	p	q	r	à	á	æ	ž	Æ	€
A-	µ	~	s	t	u	v	w	x	y	z	ı	ı	ð		þ	®
B-	¢	£	¥	·	©	§	¶	œ	ÿ	¬	š	-		Ž	×	
C-	{	A	B	C	D	E	F	G	H	I	-	ô	ö	ò	ó	õ
D-	}	J	K	L	M	N	O	P	Q	R	¹	û	ü	ù	ú	ÿ
E-	\	÷	S	T	U	V	W	X	Y	Z	²	Ô	Ö	Ò	Ó	Õ
F-	0	1	2	3	4	5	6	7	8	9	³	Û	Ü	Ù	Ú	ÿ

Représentation des caractères : Unicode

Unicode est un standard qui explique comment représenter et manipuler du texte. Il contient en particulier une liste de plus de 100000 caractères.

On trouve ensuite plusieurs façons de les représenter :

- UTF-32 : Représente chaque caractère sur 32 bits (donc 4 octets)
- UTF-8 : Représente la majorité des caractères fréquents sur 8 bits, d'autres sur 16 bits, 24 ou 32 bits

Caractère	Code UTF8
a	01100001
é	11000011 10101001
€	11100010 10000010 10101100
ℓ	11110000 10011101 10011111 10011001

Table utilisée en cours et en TD

l	00000	h	01000	p	10000	x	11000
a	00001	i	01001	q	10001	y	11001
b	00010	j	01010	r	10010	z	11010
c	00011	k	01011	s	10011	.	11011
d	00100	l	01100	t	10100	,	11100
e	00101	m	01101	u	10101	'	11101
f	00110	n	01110	v	10110	!	11110
g	00111	o	01111	w	10111	?	11111

ne permet de représenter que des minuscules et quelques signes de ponctuation, mais bien suffisant pour les exercices.

Récapitulatif

La séquence suivante :

11110000 10011101 10011111 10011001

peut donc représenter :

- Les 4 nombres 240, 157, 159, 153
- Les 2 nombres 61597 et 40857
- Le nombre 4036861849
- Les 4 caractères ≡¥fÖ (en CP437)
- Les 4 caractères 🍏ùüô (en Mac OS Roman)
- Le caractère 1 (en UTF-8)
- Les instructions assembleur x86 suivantes : LOCK POPF; LAHF;
CDQ

Un exemple

Un logiciel d'archivage permet de regrouper en un seul fichier plusieurs fichiers et répertoires afin, par exemple, de les stocker ensuite plus facilement, ou de les compresser.

On va ici archiver deux fichiers :

toto.py (16 caractères)

```
print "bonjour"
```

readme.txt (7 caractères)

```
blabla
```

avec deux logiciels différents

Un exemple : Microsoft Cabinet

```
MSCF~  
 ,♥☺☺☺☺_☺▶  
3<rN_toto.py•▶3<|N_readme.txtaC↓  
↓↓print_ "bonjour"▣blabla▣
```

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	/.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Suit ensuite une description de chaque fichier

- Taille du fichier (ici 16)
- Date et heure du fichier
- Permissions : 0x0020 signifie que le fichier est exécutable
- Nom du fichier

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Suit ensuite une description de chaque fichier

- Taille du fichier (ici 16)
- Date et heure du fichier
- Permissions : 0x0020 signifie que le fichier est exécutable
- Nom du fichier

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Suit ensuite une description de chaque fichier

- Taille du fichier (ici 16)
- Date et heure du fichier
- Permissions : 0x0020 signifie que le fichier est exécutable
- Nom du fichier

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Suit ensuite une description de chaque fichier

- Taille du fichier (ici 16)
- Date et heure du fichier
- Permissions : 0x0020 signifie que le fichier est exécutable
- Nom du fichier

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Suit ensuite une description de chaque fichier

- Taille du fichier (ici 16)
- Date et heure du fichier
- Permissions : 0x0020 signifie que le fichier est exécutable
- Nom du fichier

Un exemple : Microsoft Cabinet

00000000	534d	4643	0000	0000	007e	0000	0000	0000	MSCF.....~.....
00000016	002c	0000	0000	0000	0103	0001	0002	0000	,.....
00000032	0000	0000	005f	0000	0001	0000	0010	0000_.....
00000048	0000	0000	0000	3c33	4e72	0020	6f74	6f743<rN .toto
00000064	702e	0079	0007	0000	0010	0000	0000	3c33	.py.....3<
00000080	4e7c	0020	6572	6461	656d	742e	7478	6100	N .readme.txt.a
00000096	1943	170b	1700	7000	6972	746e	2220	6f62	C.....print "bo
00000112	6a6e	756f	2272	620a	616c	6c62	0a61		njour".blabla.

- Entête, permettant d'identifier le type du fichier (ici : une archive CAB)
- Taille de l'archive (ici 127)
- Début du premier fichier de l'archive
- Nombre de fichiers dans l'archive (ici 2)

Suit ensuite une description de chaque fichier

- Taille du fichier (ici 16)
- Date et heure du fichier
- Permissions : 0x0020 signifie que le fichier est exécutable
- Nom du fichier

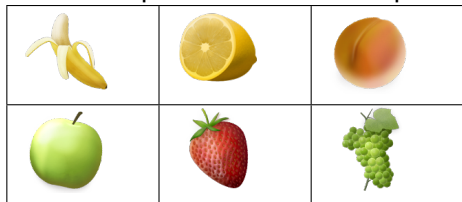
Puis le contenu de chaque fichier

Autre exemple : GNU tar

Traité en TP :

```
00000000  6f74  6f74  702e  0079  0000  0000  0000  0000  | toto.py..... |
00000016  0000  0000  0000  0000  0000  0000  0000  0000  | ..... |
...
00000096  0000  0000  3030  3030  3537  0035  3030  3130  | ...0000755.0001 |
00000112  3537  0030  3030  3130  3537  0030  3030  3030  | 750.0001750.0000 |
00000128  3030  3030  3230  0030  3131  3233  3235  3137  | 0000020.11325271 |
00000144  3332  0030  3130  3532  3430  2000  0030  0000  | 230.012504. 0... |
00000160  0000  0000  0000  0000  0000  0000  0000  0000  | ..... |
...
00000256  7500  7473  7261  2020  6500  656a  6e61  6564  | .ustar .ejeande |
00000272  006c  0000  0000  0000  0000  0000  0000  0000  | l..... |
00000288  0000  0000  0000  0000  6500  656a  6e61  6564  | .....ejeande |
00000304  006c  0000  0000  0000  0000  0000  0000  0000  | l..... |
00000320  0000  0000  0000  0000  0000  0000  0000  0000  | ..... |
...
00000512  7270  6e69  2074  6222  6e6f  6f6a  7275  0a22  | print "bonjour". |
00000528  0000  0000  0000  0000  0000  0000  0000  0000  | ..... |
...
00001024  6572  6461  656d  742e  7478  0000  0000  0000  | readme.txt..... |
00001040  0000  0000  0000  0000  0000  0000  0000  0000  | ..... |
...
00001120  0000  0000  3030  3030  3436  0034  3030  3130  | ...0000644.0001 |
00001136  3537  0030  3030  3130  3537  0030  3030  3030  | 750.0001750.0000 |
00001152  3030  3030  3030  0037  3131  3233  3235  3137  | 0000007.11325271 |
00001168  3532  0035  3130  3133  3435  2000  0030  0000  | 255.013154. 0... |
00001184  0000  0000  0000  0000  0000  0000  0000  0000  | ..... |
...
00001280  7500  7473  7261  2020  6500  656a  6e61  6564  | .ustar .ejeande |
00001296  006c  0000  0000  0000  0000  0000  0000  0000  | l..... |
00001312  0000  0000  0000  0000  6500  656a  6e61  6564  | .....ejeande |
00001328  006c  0000  0000  0000  0000  0000  0000  0000  | l..... |
00001344  0000  0000  0000  0000  0000  0000  0000  0000  | ..... |
```

- On veut représenter informatiquement une liste de fruits.




- Exemple : représenter









Dessins de Kevin Andersson - www.kevinandersson.dk

Exemple (1)

 00000000	 00000001	 00000010
 00000011	 00000100	 00000101







Exemple (1)

 00000000	 00000001	 00000010
 00000011	 00000100	 00000101







Codage :









Exemple (1)

 00000000	 00000001	 00000010
 00000011	 00000100	 00000101

Codage :

					
00000000	00000011	00000000	00000000	00000001	00000101







Exemple (1)

 00000000	 00000001	 00000010
 00000011	 00000100	 00000101

Décodage :

000000100000001100000100000000000000010100000000







Exemple (1)

 00000000	 00000001	 00000010
 00000011	 00000100	 00000101

Décodage :

00000010 00000011 00000100 00000000 00000101 00000000

Exemple (1)







 00000000	 00000001	 00000010
 00000011	 00000100	 00000101

Décodage :







00000010 00000011 00000100 00000000 00000101 00000000



Exemple (2)

 000	 001	 010
 011	 100	 101







Exemple (2)

 000	 001	 010
 011	 100	 101

Codage :



Exemple (2)

 000	 001	 010
 011	 100	 101

Codage :



000



011



000



000









001



101







Exemple (2)

 000	 001	 010
 011	 100	 101

Décodage :

010011100000101000

Exemple (2)

 000	 001	 010
 011	 100	 101

Décodage :

010

011







100

000

101

000

Exemple (2)

 000	 001	 010
 011	 100	 101

Décodage :

010



011



100



000







101









000



Exemple (3)

 000	 001	 01
 10	 110	 111





Exemple (3)

 000	 001	 01
 10	 110	 111

Codage :



Exemple (3)

 000	 001	 01
 10	 110	 111

Codage :



000



10



000



000









001



111







Exemple (3)

 000	 001	 01
 10	 110	 111

Décodage :

101100000101001

Exemple (3)

 000	 001	 01
 10	 110	 111






Décodage :

101100000101001

10



Exemple (3)

 000	 001	 01
 10	 110	 111

Décodage :

101100000101001




10



110



Exemple (3)

 000	 001	 01
 10	 110	 111

Décodage :

10110000101001

10








110



000



Exemple (3)

 000	 001	 01
 10	 110	 111

Décodage :

10110000**0**101001

10



110







000



01



Exemple (3)

 000	 001	 01
 10	 110	 111

Décodage :

101100000101001

10



110



000







01



01



Exemple (3)

 000	 001	 01
 10	 110	 111

Décodage :

101100000101001

10



110



000



01









01



001









Exemple (4)

 011	 0111	 01111
 10	 100	 1000

Décodage :

01101110011100

Exemple (4)

 011	 0111	 01111
 10	 100	 1000







Décodage :

01101110011100

011



Exemple (4)

 011	 0111	 01111
 10	 100	 1000

Décodage :

01101110011100







011



0111



Exemple (4)

 011	 0111	 01111
 10	 100	 1000

Décodage :

01101110011100







011



011



Exemple (4)

 011	 0111	 01111
 10	 100	 1000

Décodage :

01101110011100

011









011



10



Exemple (4)

 011	 0111	 01111
 10	 100	 1000

Décodage :

01101110**011**100

011



011









10



011



Exemple (4)

 011	 0111	 01111
 10	 100	 1000

Décodage :

01101110011100

011



011



10









011



100









Exemple (5)

 011	 0111	 01111
 10	 100	 0011

Décodage :

01101110011100

Exemple (5)

 011	 0111	 01111
 10	 100	 0011







Décodage :

01101110011100

011



Exemple (5)

 011	 0111	 01111
 10	 100	 0011

Décodage :

01101110011100







011



0111



Exemple (5)

 011	 0111	 01111
 10	 100	 0011

Décodage :

01101110011100

011









0111



0011



Exemple (5)

 011	 0111	 01111
 10	 100	 0011

Décodage :

01101110011100

011



0111









0011



100



Exemple (5)

 011	 0111	 01111
 10	 100	 0011

Décodage :

01101110011100

011



011



10



011



100



Définition

Un code (binaire) est une façon d'associer à chaque objet (ici des fruits) une suite de 0 et de 1 (un mot)

Définition

Un code est uniquement déchiffrable (ou non ambigu) si on peut retrouver de façon unique toute liste d'objets à partir de la suite de 0 et de 1.

Les 5 exemples précédents sont non ambigus, sauf le dernier.

Problème

Comment peut-on savoir qu'un code donné est uniquement déchiffrable ?







Problème

Comment créer un code uniquement déchiffrable ?

Code bloc

Définition

Un block code est un code où tous les mots du code sont différents et de même longueur.

 000	 101	 100
 011	 110	 111

Proposition

Un block code est non-ambigu.

Théorème

On note l_i la longueur du code pour le i -ème objet. On suppose qu'il y a n objets.

Si un code est non-ambigu, alors

$$\frac{1}{2^{l_1}} + \frac{1}{2^{l_2}} + \dots + \frac{1}{2^{l_n}} \leq 1$$

Exemple : peut-on trouver un code pour les fruits de sorte que

- 🍌 soit codé sur 1 bit ;
- 🍊, 🍋, 🍏, 🍓 soient codés sur 3 bits ;
- 🍇 sur 4 bits ?

Test de Sardinas-Patterson

Soit C le code. On le met dans une colonne C_0

A chaque étape i

- Si un mot de la première colonne C_0 commence un mot de la dernière colonne C_i (ou vice versa), on met le reste dans une nouvelle colonne C_{i+1}

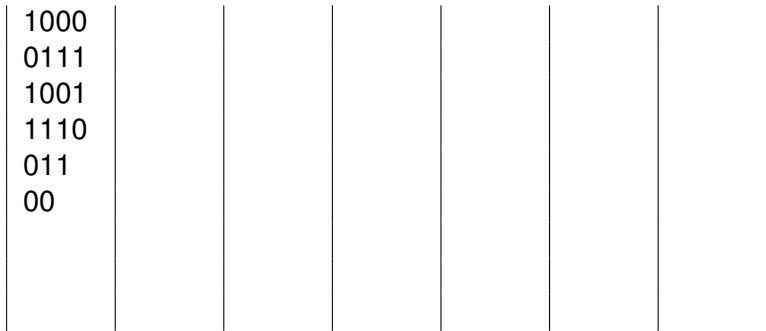
On arrête le calcul lorsqu'on boucle

Théorème

C est un code non-ambigu si et seulement si on n'atteint jamais un mot de C au cours de l'exécution

Exemple

1000
0111
1001
1110
011
00



Exemple

1000	1							
0111								
1001								
1110								
011								
00								

Exemple

1000	1	000					
0111		001					
1001		110					
1110							
011							
00							

Exemple

1000	1	000	0				
0111		001	1				
1001		110					
1110							
011							
00							

Exemple

1000	1	000	0	000			
0111		001	1	111			
1001		110		001			
1110				110			
011				11			
00				0			

Exemple

1000	1	000	0	000	111		
0111		001	1	111	0		
1001		110		001	10		
1110				110	11		
011				11	1		
00				0			

Exemple

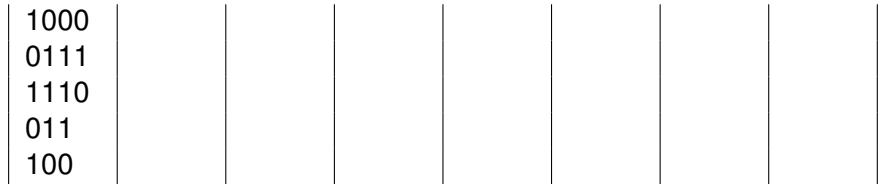
1000	1	000	0	000	111	00
0111		001	1	111	0	000
1001		110		001	10	111
1110				110	11	01
011				11	1	001
00				0		0
						10
						110
						11

Exemple

1000	1	000	0	000	111	00
0111		001	1	111	0	000
1001		110		001	10	111
1110				110	11	01
011				11	1	001
00				0		0
						10
						110
						11

C est ambigu

Exemple



Exemple

1000	0								
0111	1								
1110									
011									
100									

Exemple

1000	0	000						
0111	1	111						
1110		110						
011		11						
100		00						

Exemple

1000	0	000	0					
0111	1	111	10					
1110		110						
011		11						
100		00						

Exemple

1000	0	000	0	00				
0111	1	111	10	111				
1110		110		11				
011		11		0				
100		00						

Exemple

1000	0	000	0	00	111		
0111	1	111	10	111	0		
1110		110		11	10		
011		11		0	11		
100		00					

Exemple

1000	0	000	0	00	111	00	
0111	1	111	10	111	0	111	
1110		110		11	10	0	
011		11		0	11	10	
100		00				11	

Exemple

1000	0	000	0	00	111	00	00
0111	1	111	10	111	0	111	111
1110		110		11	10	0	0
011		11		0	11	10	10
100		00				11	11

Exemple

1000	0	000	0	00	111	00	00
0111	1	111	10	111	0	111	111
1110		110		11	10	0	0
011		11		0	11	10	10
100		00				11	11

C est non ambigu

Définition

Un code est préfixe si aucun mot n'est un préfixe d'un autre mot.

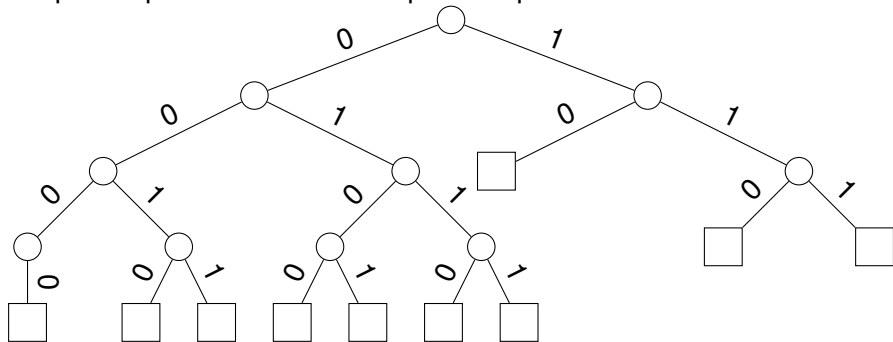
Exemple :

	0000		0101		110
	0010		0110		111
	0011		0111		
	0100		10		

Remarque : un block code est préfixe.

Code préfixe

On peut représenter les codes préfixes par des arbres :



Théorème

Un code préfixe est non-ambigu.

Preuve : pour décoder, on suit le chemin dans l'arbre. Dès qu'on arrive à une feuille, on a fini et on repart du début.

- Les codes préfixes sont *instantanés* : Dès qu'on a fini de lire le premier mot de code, on *sait* qu'on l'a lu et qu'on peut passer au deuxième.

Théorème

Soit l_i des entiers. Si les l_i vérifient

$$\frac{1}{2^{l_1}} + \frac{1}{2^{l_2}} + \dots + \frac{1}{2^{l_n}} \leq 1$$

Alors il existe un code préfixe tel que le i ème objet a pour longueur l_i .

Théorème

On note l_i la longueur du code pour le i -ème objet. Si le code est non-ambigu, alors

$$\frac{1}{2^{l_1}} + \frac{1}{2^{l_2}} + \cdots + \frac{1}{2^{l_n}} \leq 1$$

Théorème

Soit l_i des entiers. Si les l_i vérifient

$$\frac{1}{2^{l_1}} + \frac{1}{2^{l_2}} + \cdots + \frac{1}{2^{l_n}} \leq 1$$

Alors il existe un code préfixe tel que le i ème objet a pour longueur l_i .

Théorème

Si C est un code non-ambigu, on peut trouver un code préfixe avec exactement les mêmes longueurs de mot.

Les codes qui ne sont pas préfixes ne servent à rien.
Comment trouver, sachant les l_i , le code préfixe qui convient ?

Conclusion

- Pour coder des objets, on utilisera des codes non-ambigus, et souvent des codes préfixes.
- Si on sait qu'un objet apparait très souvent, il faut lui donner un code plus petit que les autres.

Comment faire ? C'est le prochain cours.