Représentation des caractères

1 Introduction

Un ordinateur ne manipule que des 0 et des 1 : comment le texte est-il codé ? Puisqu'un texte est une suite de caractères, on va s'intéresser à la représentation des caractères, c'est-à-dire au codage des lettres minuscules et capitales, des chiffres, des signes de ponctuation et autres symboles. Le principe est le suivant :

— Chaque caractère est identifié par un code unique qui est un entier naturel. La correspondance entre le caractère et son code est appelé un **Charset**.

— Le code n'étant pas utilisable tel quel par un ordinateur qui ne comprend que le binaire, il faut donc représenter les codes par des octets : c'est l'**Encoding**.

2 Le code ASCII

2.1 Présentation

Le code ASCII (*American Standard Code for Information Interchange*) définit 128 caractères numérotés de 0 à 127 et codés en binaire de 000 0000 à 111 1111. Sept bits suffisent donc. Cependant, depuis les années 1970, presque tous les ordinateurs travaillent avec des multiples de huit bits (un octet) : chaque caractère d'un texte en ASCII est souvent stocké sur huit bits, le huitième bit (celui de poids fort) étant égal à 0.

Parmi ces 128 caractères :

— 95 sont imprimables : il s'agit des chiffres de 0 à 9, des lettres minuscules et capitales de A à Z, et de symboles mathématiques et de ponctuation ;

— les 33 autres caractères (les caractères de numéro 0 à 31 et le 127) ne sont pas affichables : ils correspondent à des commandes de contrôle de terminal informatique. Par exemple, le caractère numéro 127 est la commande pour effacer. Le caractère numéro 32 est l'espace. Le caractère numéro 7 provoque l'émission d'un signal sonore.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Exercice 1

1. Quel nombre suffit-il d'ajouter au code d'une lettre capitale pour obtenir la lettre minuscule correspondante ?
2. En pratique, que suffit-il de changer à l'écriture binaire de ce code ?

On peut représenter cette table sous une forme plus condensée en ne donnant que l'Encoding (ici en hexadécimal) :

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
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	{	|	}	~	DEL

Activité 1 : taille d'un texte

1. Donner la taille (en octets) de la phrase suivante (sans les guillemets) : « La NSI, c'est chouette ! »
2. Écrire cette phrase dans un fichier `chouette` :
   - Créer un fichier `chouette` vide à l'aide de la commande `vim chouette`.
   - Passer en mode insertion en appuyant sur `i`, puis saisir la phrase « La NSI, c'est chouette ! ».
   - Appuyer sur la touche ESC pour sortir du mode insertion (on est alors dans le mode

normal de Vim).

— Sauvegarder les modifications faites en appuyant sur `:w` suivi de `Entrée`.

— Quitter Vim à l'aide de `:q` suivi de `Entrée`.

3. Quelle commande permet de connaître la taille du fichier `chouette` ? Vérifier que le résultat obtenu est cohérent avec la taille déterminée à la question 1.
4. Écrire maintenant la même phrase dans le logiciel de traitement de texte `LibreOffice Writer`.

— Quelle est la taille du fichier obtenu ?

— Quelles peuvent en être les explications ?

Activité 2 : utilisation de la table ASCII

1. Coder en binaire le mot "Orwell".
2. Voici maintenant une phrase codée en binaire :

   01010111 01100001 01110010 00100000 01101001 01110011

   00100000 01010000 01100101 01100001 01100011 01100101 00101110

   La retrouver.
3. Vérifier vos réponses sur https://mothereff.in/binary-ascii.
4. Peut-on coder en binaire la phrase « L'Histoire était récrite, mais des fragments de la littérature du passé survivraient çà et là, imparfaitement censurés et, aussi longtemps que l'on gardait l'ancilangue, il était possible de les lire. » ?

2.2 Limitations

Le code ASCII a d'abord été conçu pour des textes écrits en anglais. Il n'est pas adapté pour représenter des textes écrits dans d'autres langues, mêmes celles qui, comme le français, utilisent l'alphabet latin, car ces langues utilisent par exemple des accents, des cédilles etc. Il a donc fallu étendre la table ASCII pour pouvoir coder les nouveaux caractères, en utilisant le 8^e bit.

Exercice 2

Combien de nouvelles possibilités l'utilisation du 8^e bit engendre-t-elle ?

3 Premières extensions de la table ASCII

La norme ISO 8859-1 (appelée aussi Latin-1 ou Europe occidentale), dont le nom complet est ISO/CEI 8859-1, est une des premières extensions de la table ASCII. Elle a été conçue dans les années 1980 et permet de coder 191 caractères de l'alphabet latin qui avaient à l'époque été jugés essentiels dans l'écriture, mais omet quelques caractères fort utiles (comme la ligature œ).

Cette norme est très utilisée dans les pays occidentaux et a donné lieu à quelques extensions et adaptations dont Windows-1252 et ISO 8859-15. Ces deux extensions prennent notamment en compte la ligature œet le symbole €.

Voici deux tableaux présentant côte à côte ces deux extensions :

ISO/CEI 8859-15																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	non utilisé															
1x																
2x		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x	non utilisé															
9x																
Ax		¡	¢	£	€	¥	Š	§	š	©	ª	«	¬		®	¯
Bx	°	±	²	³	Ž	µ	¶	·	ž	¹	º	»	Œ	œ	Ÿ	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Windows-1252 (CP1252)																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8x	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ		Ž	
9x		‘	’	“	”	•	–	—	˜	™	š	›	œ		ž	Ÿ
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Exercice 3

Ces deux extensions sont-elles compatibles ?

4 Unicode

4.1 Bizarre ?

À qui n'est-il pas arrivé de lire un texte telle que celui-ci ?

> Il Ã©tait entendu que lorsque le novlangue serait une fois pour toutes adoptÃ© et que l'anci-langue serait oubliÃ©, une idÃ©e hÃ©rÃ©tique â€“ c'est-Ã -dire une idÃ©e s'Ã©cartant des principes de l'angsoc â€“ serait littÃ©ralement impensable, du moins dans la mesure oÃ¹ la pensÃ©e dÃ©pend des mots.

Bien que ceci soit de moins en moins fréquent, on trouve parfois des phrases dans lesquelles certains caractères sont remplacés par d'autres qui n'ont rien à voir et qui rendent difficiles voire empêchent la lecture et la compréhension du texte. Il s'agit ici d'un problème d'encodage et de décodage : la personne qui a écrit le texte utilise une norme différente de celle qu'utilise la personne qui le lit.

4.2 La norme Unicode

La globalisation des échanges culturels et économiques ainsi que la généralisation de l'utilisation d'Internet ont rendu nécessaire la prise en compte d'un nombre beaucoup plus important de caractères. La norme Unicode a donc été créée pour permettre des échanges de textes dans différentes langues, à un niveau mondial.

Unicode est une table de correspondance Caractère-Code (Charset) : cette table permet d'associer à chaque caractère un nom et un « numéro » unique appelé « point de code » (*code point*). Par exemple, le point de code `U+0041` est associé à la lettre `A`. Le `U+` signifie « Unicode », et la suite (0041) est un nombre écrit en hexadécimal.

Exercice 4

À quelle chaîne de caractères correspond la suite de points de code ci-dessous ?

U+0057 U+0069 U+006E U+0073 U+0074 U+006F U+006E

On pourra consulter avec profit le document « Basic Latin (ASCII) » disponible ici.

Exercice 5

Reprendre le deuxième tableau de cette fiche, et vérifier l'affirmation précédente.

Activité 3

1. En utilisant Firefox, aller sur la page internet suivante :

   `https://fr.wikipedia.org/wiki/1984_(roman)`.

2. Afficher le code source de la page (clic-droit suivi de *Code source de la page* ou avec le raccourci clavier `Ctrl+U`).
3. Quel est l'encodage défini dans l'en-tête du fichier html ?
4. Sauvegarder la page puis changer l'encodage précédent et le remplacer par `ISO-8859-1`.
5. Afficher la page modifiée à l'aide de Firefox : que constate-t-on ?

4.3 Notions d'UTF-8

Pour encoder les caractères Unicode, UTF-8 utilise un nombre variable d'octets :

1er octet	2e octet	3e octet	4e octet	Nombre de bits	Point de code
0xxxxxxx				7	jusqu'à 007F (127)
110xxxxx	10xxxxxx			$5+6=11$	de 0080 à 07FF (de 128 à 2 047)
1110xxxx	10xxxxxx	10xxxxxx		$4+6+6=16$	de 0800 à FFFF (de 2 048 à 65 535
11110xxx	10xxxxxx	10xxxxxx	10xxxxxx	$3+6+6+6=21$	de 10000 à 10FFFF (de 65 536 à 1 114 111)

Remarque 1

Le nombre de 1 consécutifs du premier octet est égal au nombre d'octets nécessaires.

Activité 4

1. Déterminer le point de code du caractère é en utilisant ce document. Combien d'octets sont nécessaires pour l'encoder en utilisant UTF-8 ? Déterminer cet encodage.
2. En quoi le codage binaire du é peut-il être décodé si on se trompe d'encodage et qu'on utilise par exemple la norme ISO 8859-1 ?
3. Vérifier votre réponse à l'aide du texte du paragraphe 4.1.
4. Il reste des caractères étranges dans le texte : "Ã¹". Comment retrouver la signification de ces caractères ?