Image et son numérique

ED 1

Exercice 1: Algorithme de compression LZW

Cette algorithme a été inventé par Jakob Ziv et Abraham Lempel (1977, 78). Il part d'un codage d'un alphabet et construit un dictionnaire de mots eux-mêmes codés.

En compression on lit les données de sorte à former des chaînes. Si une chaîne construite n'existe pas dans le dictionnaire on crée une entrée dans ce dictionnaire et on l'ajoute.
Donc chaque nouvelle chaîne ajoutée au dictionnaire est formée d'une chaîne déjà existante suivi du "caractère courant".

Le pseudo code de LZW utilisant un alphabet de type ASCII est donné par :

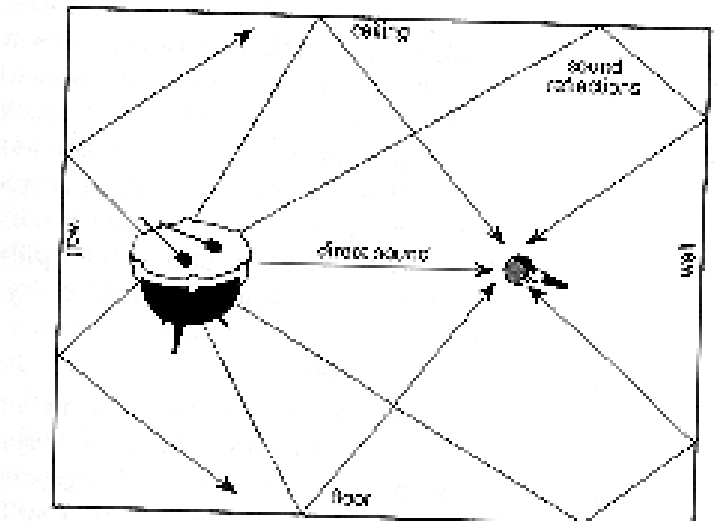
```
w : chaîne de caractère ; K caractère ;
w := NIL
tant qu'il y a des caractères
        lire un caractère K (sur 8 bits)
        si wK existe déjà dans le dictionnaire
                w := wK
        sinon
                écrire le code de w (sur 12 bits)
                ajouter wK dans le dictionnaire en créant son code
                w := K
fin tant que
écrire code de w (sur 12 bits)
```

Une définition du code ASCII est donnée en annexe. Les mots du dictionnaires qui ne sont pas des codes ASCII sont codés séquentiellement dans l'ordre de création à partir de 256.

Question 1 : Coder la Chaîne **:** /QED/QE/QEE/QEB
Question 2 : Pourquoi cette algorithme compresse t'il les données?
Question 3 : Donnez le pseudo code de l'algorithme de décompression

Exercice 2
Question 1 : Quelle sont, d'un point de vue perceptif, les grandes différences entre la perception du son et celle de l'image?

Dans une scène les aspects sonores comporte
Des sources qui peuvent être ou audio ou midi
Des caractéristiques acoustiques de l'environnement qui représentent l'effet d'absorption ou de réflexion du son sur les objets
Un auditeur (micro virtuel)



Question 2 En cours un pipe-line graphique a été présenté. Proposer un schéma de pipe line sonore.

The standard ASCII code defines 128 character codes (from 0 to 127), of which, the first 32 are control codes (non-printable), and the other 96 are representable characters:

*	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	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
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	{	|	}	~	

* This panel is organized to be easily read in hexadecimal: row numbers represent the first digit and the column numbers represent the second one. For example, the **`A`** character is located at the **`4th`**row and the **`1st`** column, for that it would be represented in hexadecimal as **`0x41`** (**`65`**).

In addition to the 128 standard ASCII codes there are other 128 that are known as extended ASCII, and that are platform-dependent. So there is more than one extended ASCII character set.

The two most used extended ASCII character sets are the one known as OEM, that comes from the default character set incorporated by default in the IBM-PC and the other is the ANSI extend ASCII which is used by recent operating systems.

First of them, the OEM character set, is included in the immense majority of PC compatibles when starting before loading any operating system or under MS-DOS system. It includes diverse foreign signs, some marked characters and pieces to represent panels. Unfortunately it is usually redefined according to regional configurations to incorporate own symbols in many countries.

OEM Extended ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	ø	ε	∩
F	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	