



DIRO
IFT 1215

INTRODUCTION AUX SYSTÈMES INFORMATIQUES

FORMAT DES DONNÉES

Max Mignotte

Département d'Informatique et de Recherche Opérationnelle
Http: [//www.iro.umontreal.ca/~mignotte/](http://www.iro.umontreal.ca/~mignotte/)
E-mail: mignotte@iro.umontreal.ca

FORMAT DES DONNÉES

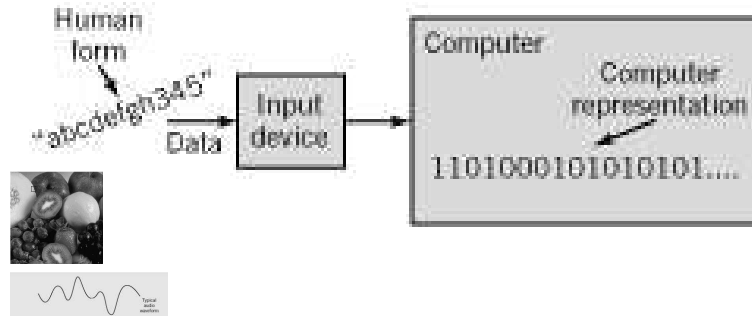
SOMMAIRE

Introduction	2
Format Propriétaire -Standard	3
Code Alphanumérique	4
Entrée Alphanumérique	9
Format d'Images	11
Format Audio et Vidéo	16
Compression de données	18
Format de Données interne	21

FORMAT DES DONNÉES

INTRODUCTION

Introduction



- Les données (texte, nb, images, etc.) dans l'ordinateur ne peuvent être représenté que par des 0 ou 1
- Les données d'entrées doivent être convertit dans un format approprié qui permettent à l'ordinateur de les
 - ▷ Stocker
 - ▷ Transmettre
 - ▷ Reconnaître
 - ▷ Traiter
- Les données d'entrées peuvent être continue (son, image analogique) ou discrète (caractère, etc)
- Le plus souvent, l'ordinateur stocke (au début de chaque fichier de donnée) de l'information qui décrit la signification des données ▷ *metadata*.

FORMAT DES DONNÉES

FORMAT PROPRIÉTAIRE - STANDARD

Format Propriétaire

Unique à un produit ou une compagnie
(*Exemples: Microsoft Word, Corel Word Perfect, etc.*).

Standard

–1– Un format propriétaire peut devenir un standard
(e.g., PDF, Postscript, etc.)

–2– Un comité d'expert est constitué pour résoudre
un problème et proposer un standard pour un problème
particulier

- ▷ ISO -International Standards Organization
- ▷ CSA -Canadian Standards Association
- ▷ ANSI -American National Standards Institute
- ▷ IEEE -Institute for Electrical and Electronics Engineers

Type of data	Standard(s)
Alphanumeric	Unicode, ASCII, EBCDIC
Image (bitmap)	GIF (graphical image format), TIFF (tagged image file format), PNG (portable network graphics)
Image (object)	PostScript, JPEG, SWF (Macromedia Flash), SVG
Outline graphics and fonts	PostScript, TrueType
Sound	WAV, AVI, MP3, MIDI, WMA
Page description	pdf (Adobe Portable Document Format), HTML, XML
Video	Quicktime, MPEG-2, RealVideo, WMV

FORMAT DES DONNÉES

CODE ALPHANUMÉRIQUE

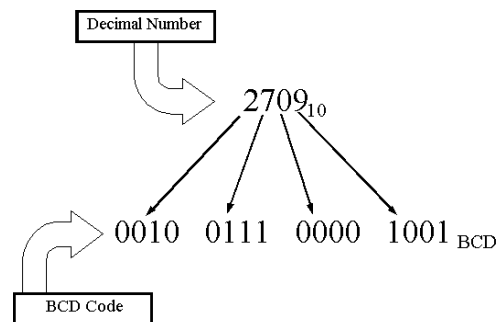
Caractères Alphanumériques

- Lettres de l'alphabet (minuscule et majuscules)
- Les caractères : 1, 2, 3, 4, ...etc.
- Ponctuations : !, ?, ", (, etc.
- Caractères spéciales : *, \$, >, etc.

Quatre standards utilisés pour les coder en binaires

1. BCD (*Binary Coded Decimal*)
2. ASCII (*American Standard Code for Information Interchange*)
3. Unicode
4. EBCDIC (*Extended Binary Coded Decimal Interchange*)

BCD



ASCII

- Développé initialement par le *American National Standards Institute (ANSI)*
- Code de 7 bits (128 entrées possibles, 95 graphiques et 33 de contrôle), stocké sur un Octet [*Byte*]
- Le 8 ième bit est quelquefois inutilisé, utilisé comme bit de parité, ou pour codé 128 autres symboles

FORMAT DES DONNÉES

CODE ALPHANUMÉRIQUE

Table de Codage ASCII

←	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	⊙	P	'	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	~
E	SO	RS	.	>	N	^	n	_
F	SI	US	/	?	O	_	o	DEL

G à le code 47_{16} ou $0100\ 0111_2$

- 95 codes *graphiques* de 20_{16} à $7E_{16}$
 - codes alphabétiques
 - codes numériques
 - codes de ponctuation
- 35 codes *de contrôle* de 00_{16} à $1F_{16}$ et $7F$
- Latin-I ASCII; variante incluant un ensemble de caractères accentués et spéciales

Exemples:

La chaîne de caractère Hello, world !, à pour code (en hexadécimal),

48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21

FORMAT DES DONNÉES

CODE ALPHANUMÉRIQUE

Caractères graphiques

- **a** à le code hexadécimal 61₁₆. Pour convertir ce caractère en caractère majuscule (i.e., **A**), on doit soustraire au code 20₁₆ (touche *shift*)
- L'ordre des lettres est respecté (classement par ordre alphabétique par simple algorithme de trie)
- Le caractère **5** codé par le code 35₁₆ est différent du nombre 5. Pour convertir le caractère en nombres on doit soustraire au code la valeur 30₁₆.

Caractère de Contrôle

NUL	(Null) No character; used to fill space	DLE	(Data Link Escape) Similar to escape, but used to change meaning of data control characters; used to permit sending of data characters with any bit combination
SOH	(Start of Heading) Indicates start of a header used during transmission	DC1, DC2, DC3, DC4	(Device Controls) Used for the control of devices or special terminal features
STX	(Start of Text) Indicates start of text during transmission	NAK	(Negative Acknowledgment) Opposite of ACK
ETX	(End of Text) Similar to above	SYN	(Synchronous) Used to synchronize a synchronous transmission system
EOT	(End of Transmission)	STB	(End of Transmission Block) Indicates end of a block of transmitted data
ENQ	(Enquiry) A request for response from a remote station; the response is usually an identification	CAN	(Cancel) Cancel previous data
ACK	(Acknowledge) A character sent by a receiving device as an affirmative response to a query by a sender	EM	(End of Medium) Indicates the physical end of a medium such as tape
BEL	(Bell) Rings a bell	SUB	(Substitute) Substitute a character for one sent in error
BS	(Backspace)	ESC	(Escape) Provides extensions to the code by changing the meaning of a specified number of contiguous following characters
HT	(Horizontal Tab)	FS, GS, RS, US	(File, group, record, and united separators) Used in optional way by systems to provide separations within a data set
LF	(Line Feed)	DEL	(Delete) Delete current character
VT	(Vertical Tab)		
FF	(Form Feed) Moves cursor to the starting position of the next page, form, or screen		
CR	(Carriage return)		
SO	(Shift Out) Shift to an alternative character set until Si is encountered		
SI	(Shift In) see above		

FORMAT DES DONNÉES

CODE ALPHANUMÉRIQUE

Code EBCDIC

	0	1	2	3	4	5	6	7
0	NUL	DLE	DS		space	&	-	
1	SOH	DC1	SOS		RSP		/	
2	STX	DC2	FS	SYN				
3	ETX	DC3	WUS	IR				
4	SEL	ENP	BYP/INP	PP				
5	HT	NL	LF	TRN				
6	RNL	BS	ETB	NBS				
7	DEL	POC	ESC	EOT				
8	GE	CAN	SA	SBS				
9	SPS	EM	SFE	IT				
A	RPT	UB5	SM/SW	RFF	¢	!		:
B	VT	CU1	CSP	CU3	.	\$,	#
C	FF	IFS	MFA	DC4	<	*	%	⊗
D	CR	IGS	ENQ	NAK	()	~	'
E	SO	IRS	ACK		+	:	>	=
F	SI	IUS	BEL	SUB	:	~	?	*

	8	9	A	B	C	D	E	F
0					()	\	0
1	a	j	_		A	J	NSP	1
2	b	k	s		B	K	S	2
3	c	l	t		C	L	T	3
4	d	m	u		D	M	U	4
5	e	n	v		E	N	V	5
6	f	o	w		F	O	W	6
7	g	p	x		G	P	X	7
8	h	q	y		H	Q	Y	8
9	i	r	z		I	R	Z	9
A					5HY			
B								
C								
D								
E								
F								E0

- Pas de caractères pourtant très utile aujourd'hui !
comme `[]` (langage C, C++, java, fortran, etc.), `{ }` (langage C, C++), `~` (Unix, Internet, etc.), etc.
- Code 8 bits, Inventé par IBM, désuet mais beaucoup d'archives l'utilisent

FORMAT DES DONNÉES

CODE ALPHANUMÉRIQUE

Codage Unicode

Code range (in hexadecimal)	
0000–	0000–00FF Latin-1 (ASCII)
1000–	General character alphabets: Latin, Cyrillic, Greek, Hebrew, Arabic, Thai, etc.
2000–	Symbols and dingbats: punctuation, math, technical, geometric shapes, etc.
3000–	3000–33FF Miscellaneous punctuations, symbols, and phonetics for Chinese, Japanese, and Korean
4000–	Unassigned
5000–	• • • 4E00–9FFF Chinese, Japanese, Korean ideographs
A000–	Unassigned
B000–	
C000–	AC00–D7AF Korean Hanguel syllables
D000–	
E000–	Space for surrogates
F000–	E000–F8FF Private use
FFFF–	Various special characters

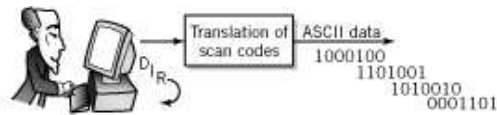
- Code de 16 bits (65 536 entrées possibles mais contient jusqu'à maintenant 38 887 caractères distincts), stocké sur deux octets [*word*]
- Le code ASCII latin-1 est englobé dans ce code
- Code multilingues: Lettres et idéogramme (Amérique, Europe, Afrique, Asie, etc.)

<http://www.unicode.org>

FORMAT DES DONNÉES

ENTRÉE ALPHANUMÉRIQUE

Du Clavier Au Binaire



- Les circuits du clavier génère un code [*scan code*] lorsque la touche est pressée et un autre lorsque la touche est libérée
- Convertit ensuite en code ASCII, EBCDIC, Unicode par **conversion logiciel** dans le PC
 - ▷ Adapté à différents langages ou claviers
 - ▷ Multiple combinaisons possibles (shift, control, etc.)
- Les caractères graphiques et de contrôle sont traités comme un flot de données et stocké dans un buffer

D'autres sources d'entrées alphanumériques

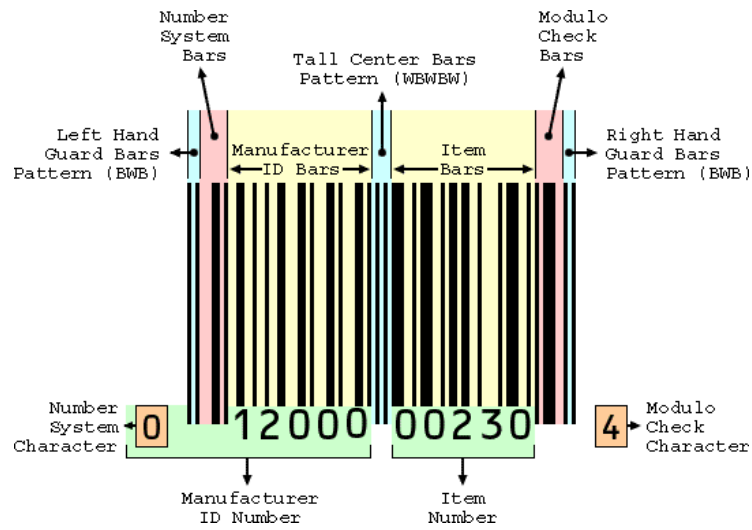
Des données alphanumériques peuvent être entrées dans l'ordinateur en utilisant

- ▷ Un Scanner et un logiciel OCR



FORMAT DES DONNÉES ENTRÉE ALPHANUMÉRIQUE

▷ Un lecteur de code bar



<http://www.digital.net/barcoder/barcode.html>

▷ Un lecteur de bande magnétique



▷ Convertisseur de signal vocal

▷ **Appareil de pointage** Exemples: souris, crayon optiques, etc.



FORMAT DES DONNÉES

FORMAT D' IMAGES

Format d'Images : 2 catégories

Produit par un scanner, caméra, logiciel de dessin, etc.

- **Images bitmap** [*raster images*]

Désigne un format de donnée qui va représenter et stocker chaque point de l'image individuellement (niveaux de gris ou niveaux de rouge, vert, bleu)

- Format gif, jpeg, pgm etc.

- **Images Vectorielle** [*object images*] [*vector images*]

Désigne un format de donnée où l'image entière est décrite par un ensemble de forme géométrique (lignes, courbes, cercles, ellipse, etc.) dont les paramètres seront ensuite stockés.

- Postscript (PS), PDF etc.

Pour les deux catégories, des différences existent et incluent

- ▷ Qualité de l'image
- ▷ Espace de stockage nécessaire
- ▷ Facilité de manipulation

FORMAT DES DONNÉES

FORMAT D' IMAGES

Image Numérique Bitmap



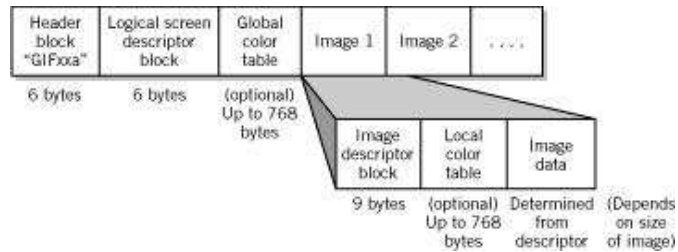
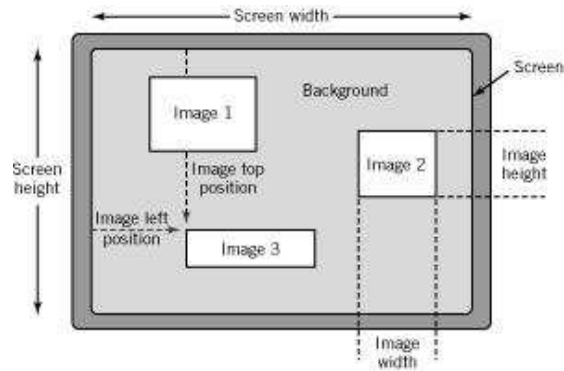
x =	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
y =															
41	210	209	204	202	197	247	143	71	64	80	84	54	54	57	58
42	206	196	203	197	195	210	207	56	63	58	53	53	61	62	51
43	201	207	192	201	198	213	156	69	65	57	55	52	53	60	50
44	216	206	211	193	202	207	208	57	69	60	55	77	49	62	61
45	221	206	211	194	196	197	220	56	63	60	55	46	97	58	106
46	209	214	224	199	194	193	204	173	64	60	59	51	62	56	48
47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	55
48	214	215	215	207	208	180	172	188	69	72	55	49	56	52	56
49	209	205	214	205	204	196	187	196	86	62	66	87	57	60	48
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

- **Image Numérique en niveau de gris:** Matrice où la valeur de chaque élément (pixel) représente l'intensité *discrète* de la lumière au point (x, y) (0 : noir, 255 : blanc pour un codage des niveaux de gris sur 8 bits [1 Byte])
- Stockage pour une image de 256×256 pixels :
 $256 \times 256 = 65536 = 64K$ Bytes
Pour réduire le stockage ▷ compression

FORMAT DES DONNÉES

FORMAT D' IMAGES

Format GIF



- Développé par CompuServe (1987)
- GIF₈₉ permet l'animation d'images
- nb couleurs : 256
- Compression sans perte, algorithme LZW (LempelZiv & Welch)

FORMAT DES DONNÉES

FORMAT D' IMAGES

Image Numérique Vectorielle

L'image est décomposé en formes géométriques (lignes, courbes, etc.) et des instructions spécifiant chacune de ses formes est stocké dans un fichier


```

288 396 translate % move origin to center of page
0 0 144 0 360 arc % define 2" radius black circle
fill

0.5 setgray % define 1" radius gray circle
0 0 72 0 360 arc
fill

0 setgray % reset color to black
-216 -180 moveto % start at lower left corner
0 360 rmoveto % and define rectangle
432 0 rmoveto % ...one line at a time
0 -360 rmoveto
closepath % completes rectangle
stroke % draw outline instead of fill

showpage % produce the image
    
```



```


% procedure to draw pie slice
%arguments graylevel, start angle, finish angle
/wedge [
  0 0 moveto
  setgray
  /angle1 exch def
  /angle2 exch def
  0 0 144 angle1 angle2 arc
  0 0 lineto
  closepath ] def

% add text to drawing
0 setgray
144 144 moveto
(baseball cards) show
-30 200 (cash) show
-216 108 (stocks) show
32 scalefont
(Personal Assets) show

showpage

%set up text font for printing
/Helvetica-Bold findfont
16 scalefont
setfont

.4 72 108 wedge fill % 108-72 = 36 = .1 circle
.8 108 360 wedge fill % 70%
% print wedge in three parts
32 12 translate
0 0 72 wedge fill
gsave
-8 8 translate
1 0 72 wedge fill
0 setgray stroke
grestore
    
```



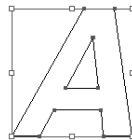
FORMAT DES DONNÉES

FORMAT D' IMAGES

- Stockage : dépend de la complexité de l'image
- Basé sur des formules mathématiques : L'image peut être facilement tournée, agrandi, sans perte de qualité (très utilisé pour les fontes de caractères)



Bitmap



Vectorielle

- Ne peut être affiché directement (excepté avec une table traçantes), nécessite de convertir l'image vectorielle en bitmap comme une imprimante postscript
- [*Page Description Language*] : nom de la liste des procédures qui décrit chacun des objets géométrique de l'image
 - ▷ Stocké en ASCII ou Unicode
 - ▷ Convertit par un programme interpréter

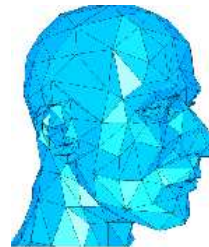
Exemples d'images vectorielles



Shreck



Toy story



FORMAT DES DONNÉES

FORMAT VIDÉO ET AUDIO

Séquence Vidéo

- Demande une grande capacité de stockage

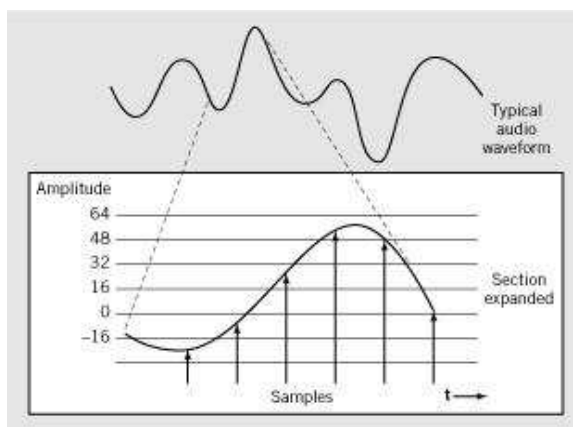
Exemples : Caméra vidéo produit de images 480×680 ,
3 Octet par pixel, 30 images par seconde

▷ 27,65 MegaOctet par seconde
(1 minute ▷ 1.6 GigaOctets)

- [Streaming Video] Séquence vidéo télécharger en temps réelle (video-conférence)
- Compression possible (Exemples : *Quicktime* d'Apple, *Windows Media Format* de Microsoft, MPEG-2)

Donné Audio

- Signal analogique digitalisé pour le convertir en numérique (AD convertisseur)



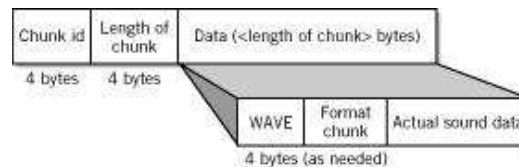
- Échantillonnage de 44.1 KHz pour un CD audio

FORMAT DES DONNÉES

FORMAT AUDIO

Format .WAV

- Inventé par Microsoft. Échantillon de son sur 8, 16 bits à une fréquence d'échantillonnage de 11.025 KHz, 22.05 KHz, 44.1 KHz en mono ou stéréo (2 × 16 bits)



Autres Formats

Format MIDI

- MIDI : *Musical Instrument Digital Interface*
- Utilisé par les compositeurs musiciens, les professionnels du son et de l'acoustique
- Instructions permettant de recréer et synthétiser de nouveaux sons et d'interfacer avec des synthétiseurs (mais ne permet pas de recréer efficacement de la voix humaine)
- 3 minutes de son \approx 10 KiloBytes

Format MP3

- Dérive du format MPEG-2 (*Moving Picture Expert Group*)
- Compression avec perte
- 3 minutes de musique \approx 2 MegaBytes

FORMAT DES DONNÉES

COMPRESSION DE DONNÉES

Compression des Données

But : Recoder les données de telle façon qu'elles ne nécessitent moins d'octets pour le stockage

- ▶ Réduction du coût de stockage
- ▶ Transmission rapide des données

- Compression avec (ex: jpeg, mpeg2, mp3, etc.) ou sans perte (l'algorithme inverse restaure les données dans sa forme originale sans altération) (ex: gif, pcx, tiff, etc.)

Taux de compression

$$C = \frac{\text{Nb. de bits après compression}}{\text{Nb. de bits avant compression}}$$

Ex: Compression avec un facteur de 10 : 1

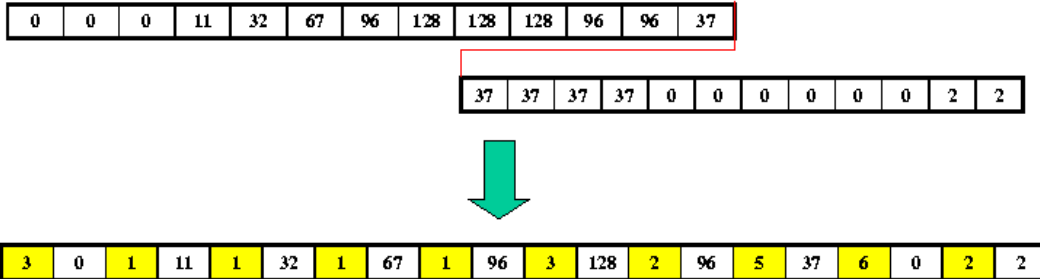
Exemple de méthode simple de compression

- Méthode RLE
- Méthode du dictionnaire

FORMAT DES DONNÉES

COMPRESSION DE DONNÉES

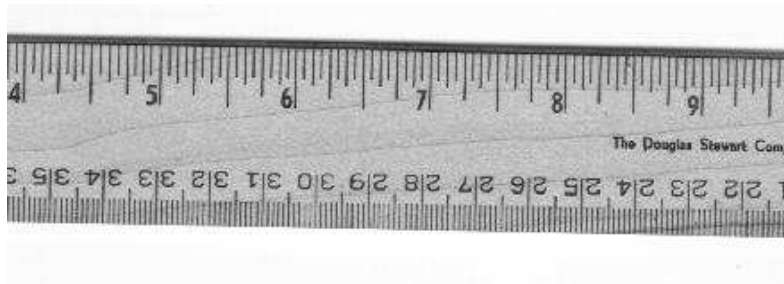
Compression RLE



Principe

- RLE: Run Length Encoding
- Création d'une nouvelle séquence dans laquelle le deuxième élément correspond au niveau de gris et le premier élément correspond au nombre de pixels consécutif possédant ce niveau de gris
- On code séparément le niveau de gris et l'occurrence de chaque pixel

▼
Fort taux de compression pour des images possédant de nb. zones de régions homogènes



FORMAT DES DONNÉES COMPRESSION DE DONNÉES

Compression avec Dictionnaire

Exemple:

“Peter Piper picked a peck of pickled peppers”

[Pe] t [er] [Pi] p [er] [pi] [ck] [ed] a [pe] [ck] of
[pi] [ck] l [ed] [pe] pp [er]s

En utilisant le dictionnaire suivant

[Pe:▲] [pi:▼] [ed : ◆] [er: ★] [ck : ►] [ck : ✕] [pe: ✓]
[Pi : □]

Et on transmet le dictionnaire et la phrase

▲t★ □p★ ▼✕◆ a ✓ ► of ▼ ✕◆ ✓pp★s

FORMAT DES DONNÉES

FORMAT DE DONNÉES INTERNE

- Toute les données sont stockées sous forme binaire de taille différentes
- Ces données peuvent être interprété pour représenter des donnée de différents type et format *via* un langage de programmation

Float, char, boolean, int, etc.

Exemple: Programme en Langage Fortran

```
//VARIABLES USED
key CHARACTER;
number INTEGER;
error stop BOOLEAN;

[
  stop = false;
  error = false;
  ReadAKey;
  WHILE NOT stop AND NOT error [
    number = 10 * number + (ASCII VALUE(key)- 48);
    ReadAKey;
  ];
  IF error
    PRINTOUT ('Illegal Character in Input')
  ELSE PRINTOUT ('Input number is `',number);
];

PROCEDURE ReadAKey [
  READ (key);
  IF (ASCII VALUE(key)=13 or ASCII VALUE(key)=32 or
    ASCII VALUE(key)=44)
    stop = TRUE;
  ELSE IF (key < '0') or (key > '9')
    error = TRUE;
];
```