



# MODULE INF112

---

TD 5  
2012 - 2013



# Plan

---

1. Préparation TP5
  - Suite de Lucas (Examen 2006-07)
  - Équation logistique (sous Excel)
  - Analyse de la séquence d'ADN
2. Structuration et codage de l'information



## 1.Préparation TP5

# Suite de Lucas

On souhaite produire une action inscrivant les 150 premières valeurs de la suite de Lucas dans la troisième colonne d'un tableau comme Excel.

- Cette suite est définie par l'équation suivante :
  - $U_1 = 0$
  - $U_2 = 3$
  - $U_n = U_{n-1} + U_{n-2}$
- On rappelle que pour écrire la valeur « 0 » dans la cellule C1, on écrit:  $\text{Cellule}(1,3) \leftarrow 0$



## 1.Préparation TP5

# Suite de Lucas

---

- Proposer un algorithme
- Paramétrer les valeurs initiales de la suite, pour calculer toutes les suites de la forme de la suite de Lucas
- Proposer un algorithme pour calculer Fibonacci à partir de Lucas Paramétré



## 1.Préparation TP5

# Lucas & Fibonacci

- Définitions

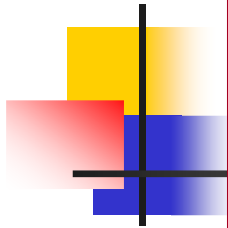
<b>Fibonacci</b>	<b>Lucas</b>
$F(n+2) = F(n+1) + F(n)$ avec $F(1) = F(2) = 1$	$L(n+2) = L(n+1) + L(n)$ avec $L(1) = 1$ and $L(2) = 3$

- Listes

<i>N</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
<b>Fibonacci</b>	1	1	2	3	5	8	13	21	34	55	89	144
<b>Lucas</b>	1	3	4	7	11	18	29	47	76	123	199	322

- Relations

$F(n+1) + F(n-1) = L(n)$	$L(n+1) + L(n-1) = F(n)*5$
--------------------------	----------------------------



Action **SuiteLucasParam**(u1,u2,c : entier)

Début

i : entier

Cellule(1,c)  $\leftarrow$  u1

Cellule(2,c)  $\leftarrow$  u2

**Pour** i = 3 à 150

Cellule(i,c) = Cellule(i-1,c) + Cellule(i-2,c)

**Fin pour**

Fin

Action FiboAPartirDeLucas

Début

**SuiteLucasParam**(0,1,3)

Fin



# Plan

---

1. Préparation TP5
  - Suite de Lucas (Examen 2006-07)
  - Équation logistique (sous Excel)
  - Analyse de la séquence d'ADN
2. Structuration et codage de l'information



## 2. Préparation du TP5

# Equation logistique : flux et reflux de la vie

- Un modèle d'évolution de populations
- Modélisation :

$$P_{t+1} = P_t + (F\_croiss * P_t * (1 - P_t))$$

**$P_t$**  «*population ancienne*» : population au temps  $t$ ,

**$P_{t+1}$**  «*population nouvelle*» : population au temps  $t+1$ .

$P_t$  est exprimé en fréquence et varie dans l'intervalle  $[0,1]$ .





## 2. Préparation du TP5

# Equation logistique

---

- Les résultats dépendent des conditions initiales :
  - de la population au temps  $t_0$
  - du taux de croissance
- Problématique
  - observer l'évolution de la population  $P_t$ ,
  - en faisant varier les conditions initiales



## 2. Préparation du TP5

# Réaliser des macros

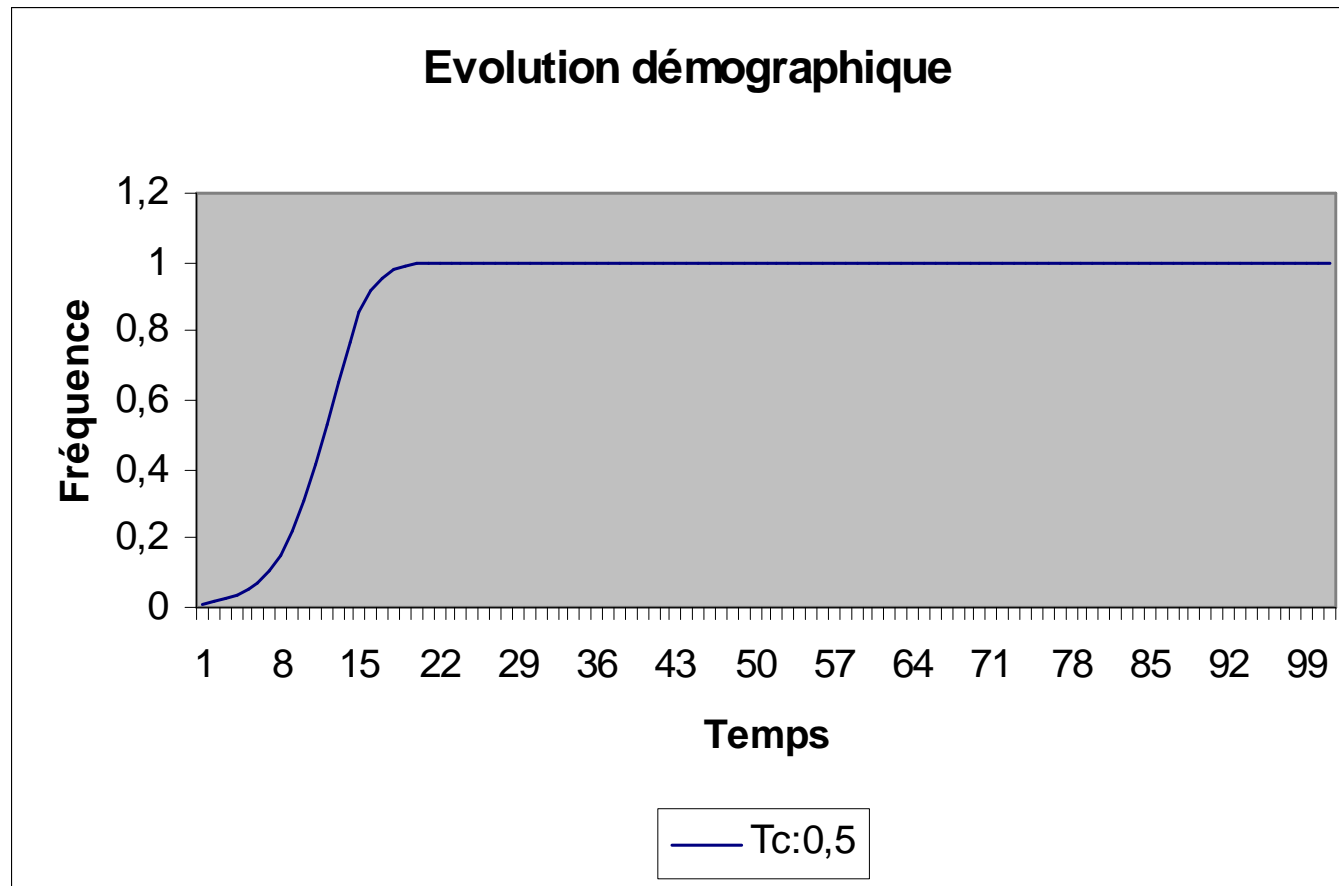
- pour remplir plusieurs tableaux de données correspondants aux différentes conditions initiales testées
- pour afficher les graphes correspondants

	A	B	C	D	E
1				Taux de croissance	0,5
2					
3	Temps	Population			
4	0	0,01			
5	1	0,01495			
6	2	0,02231325			
...	...	...	...	...	...
103	99	1			
104	100	1			



## 2. Préparation du TP5

# Afficher les résultats





## 2. Préparation du TP5

# Mode opératoire

---

- Création d'une macro DATA pour remplir une feuille de calcul
- Observation d'une macro GRAPHE pour tracer la courbe
- Itération pour exécuter le corps de la macro DATA sur 6 pages différentes en variant le taux de croissance
- Paramétrisation de GRAPHE
- Ajout de l'appel à GRAPHE dans DATA



# Plan

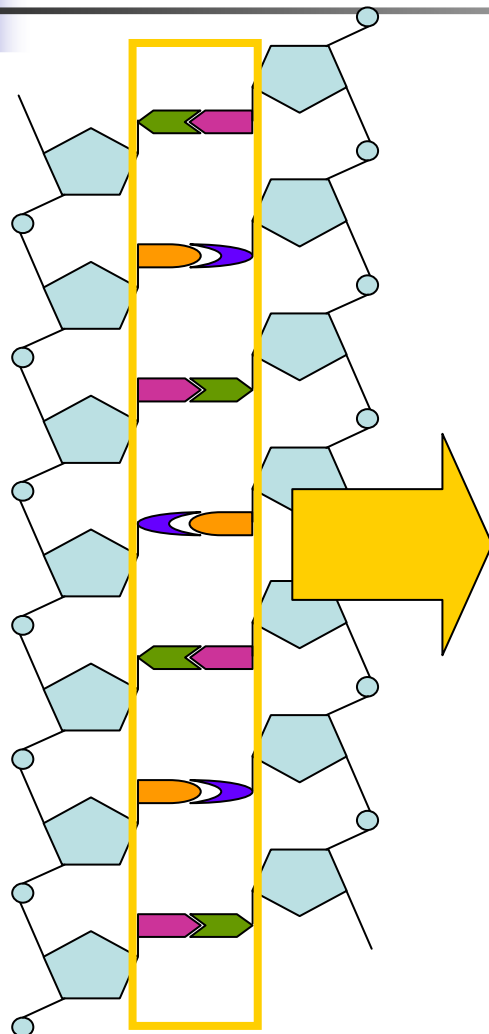
---

1. Préparation TP5
  - Suite de Lucas (Examen 2006-07)
  - Équation logistique (sous Excel)
  - Analyse de la séquence d'ADN
2. Structuration et codage de l'information



### 3. Préparation du TP5

# Analyse séquence ADN



**Séquence mono-brin :**  
**Fichier ADN.doc**  
⇒ **Transformation en ADN.txt**

```
TTCTTCGACCTCCCC  
ACCCCTTTGCGCTTT  
GAAAGCCCTCTGCCG  
CATACGAAGGCTAAC  
CCCTGAAGATGCAAA  
GTTGAATGATAACTA  
TTACTTCACAAAGCC  
ATACATGCACCTAGT  
AACGCCCGCATCTGT  
CACCGCACTTAGCAC
```

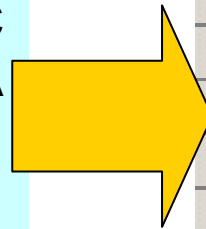


### 3. Préparation du TP5

# Analyse séquence ADN

⇒ **Importation de ADN.txt sous Excel**

```
TTCTTCGACCTCCCC
ACCCCTTTCGCCTTT
GAAAGCCCTCTGCCG
CATACGAAGGCTAAC
CCCTGAAGATGCAAA
GTTGAATGATAACTA
TTACTTCAAAAGCC
ATACATGCACCTAGT
AACGCCCGCATCTGT
CACCGCACTTAGCAC
```



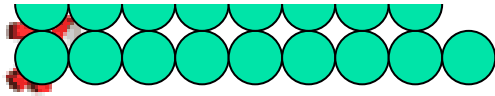
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	T	T	C	T	T	C	G	A	C	C	T	C	C	C	C
2	A	C	C	C	C	T	T	T	C	G	C	C	T	T	T
3	G	A	A	A	G	C	C	C	T	C	T	G	C	C	G
4	C	A	T	A	C	G	A	A	G	G	C	T	A	A	C
5	C	C	C	T	G	A	A	G	A	T	G	C	A	A	A
6	G	T	T	G	A	A	T	G	A	T	A	A	C	T	A
7	T	T	A	C	T	T	C	A	C	A	A	A	G	C	C
8	A	T	A	C	A	T	G	C	A	C	C	T	A	G	T
9	A	A	C	G	C	C	C	G	C	A	T	C	T	G	T
10	C	A	C	C	G	C	A	C	T	T	A	G	C	A	C

**Problématique :**

**compter les A, les T, les C et les G**

**calculer les proportions de A, de T, de C et de G**

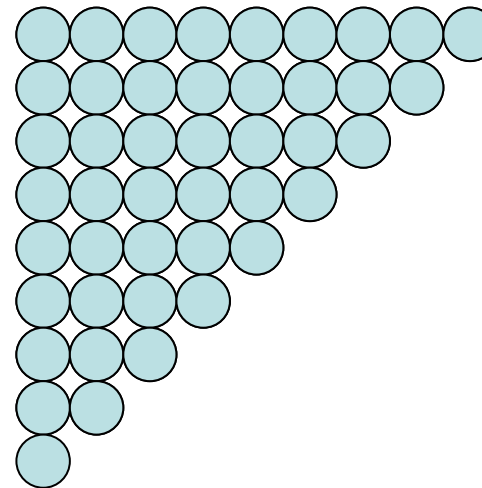
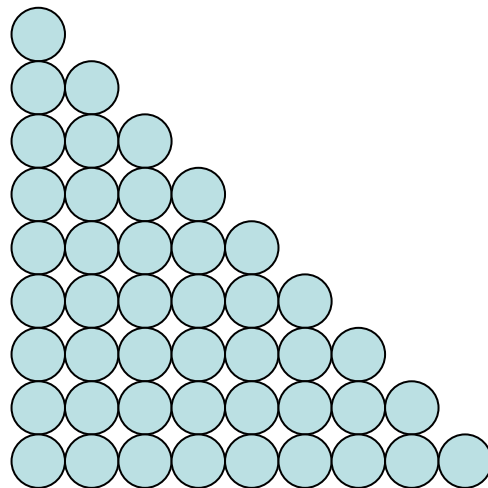
**Quel algorithme ?**



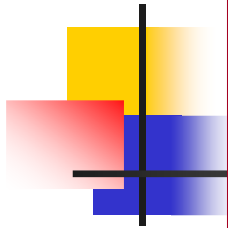
### 3. Préparation du TP5

## Autre exercice

- Proposer des algorithmes pour la réalisation de ces deux dessins







## Action PavageCercleG()

Début

$i, j$  : entier {compteurs}

$x, y$  : entier {coin supérieur gauche}

$c$  : entier {diamètre}

$x \leftarrow 100$

$y \leftarrow 350$

$c \leftarrow 20$

Pour  $i = 0$  à  $8$  {ligne}

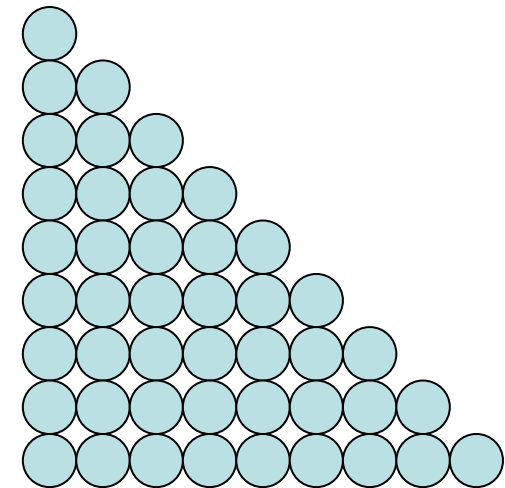
    Pour  $j = 0$  à  $8-i$  {colonne}

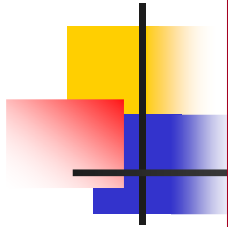
        Cercle( $x + j*c, y - i*c, c$ )

    Fin pour

Fin pour

Fin





## Action PavageCercleD()

Début

$i, j$  : entier {compteurs}

$x, y$  : entier {coin supérieur gauche}

$c$  : entier {diamètre}

$x \leftarrow 100$

$y \leftarrow 350$

$c \leftarrow 20$

Pour  $i = 1$  à  $9$  {ligne}

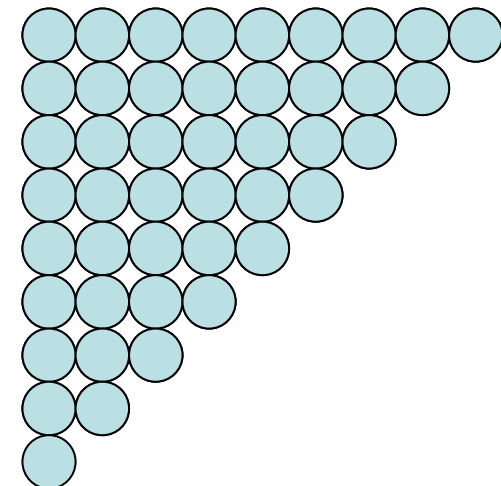
    Pour  $j = 0$  à  $i$  {colonne}

        Cercle( $x + j*c, y - i*c, c$ )

    Fin pour

Fin pour

Fin





# Plan

---

1. Préparation TP5
  - Suite de Lucas (Examen 2006-07)
  - Équation logistique (sous Excel)
  - Analyse de la séquence d'ADN
2. Structuration et codage de l'information



## 2. Structuration et codage de l'information

# Dans la vie de tous les jours

---

## Information écrite

- Via un système de codage
  - Symboles
    - Lettres (A..Z), Hiéroglyphes ...
    - Kanji, Hiragana, Katakana ...
  - Règles de composition (orthographe, grammaire)
- Structurée
- Sur un support (feuilles, livres, murs, ...)



## 2. Structuration et codage de l'information

# Pour l'ordinateur

---

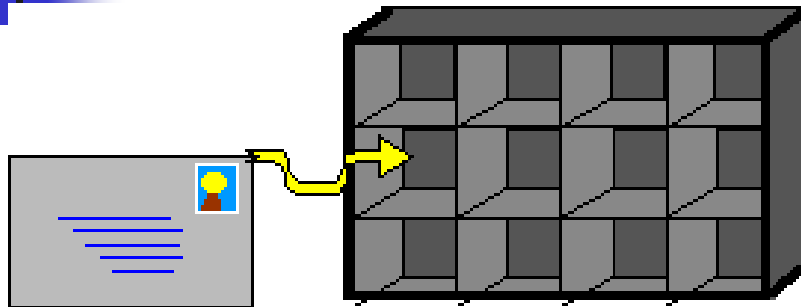
Très similaire à l'exemple précédent

- L'information est codée à partir de "bit"
  - 2 valeurs (vrai / faux ; 0 / 1; ...)
  - 8 bits = un octet
- On peut associer un sens aux suites de bits
  - Plusieurs codages possibles
- L'information est structurée
  - Fichiers, Dossiers / répertoires
- Stockée sur un support
  - disques, disquettes, clef USB, bandes magnétiques



## 2. Structuration et codage de l'information

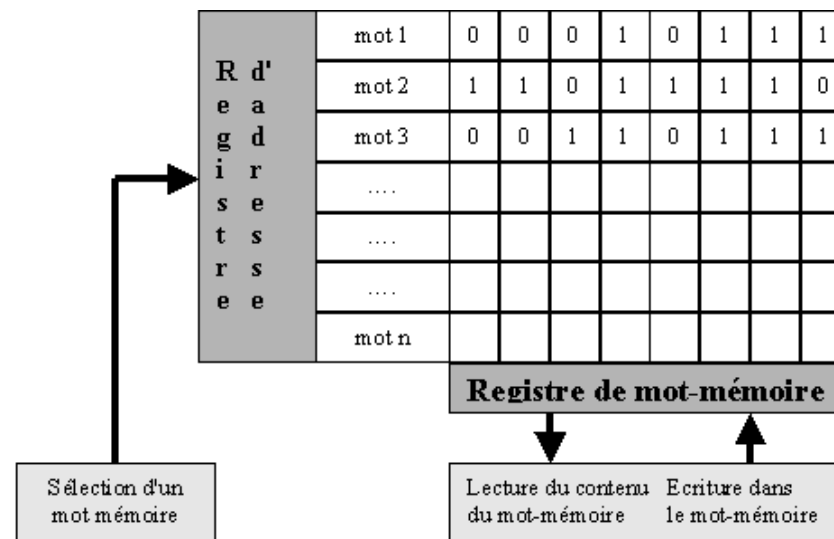
# Information et mémoire



Mémoire :

« casier avec une adresse »  
dans lequel est « rangée »  
de l'information

8 bits = octet ou byte





## 2. Structuration et codage de l'information

# Capacité mémoire

---

1 KiloOctets =  $2^{10}$  Octets = 1024 Octets

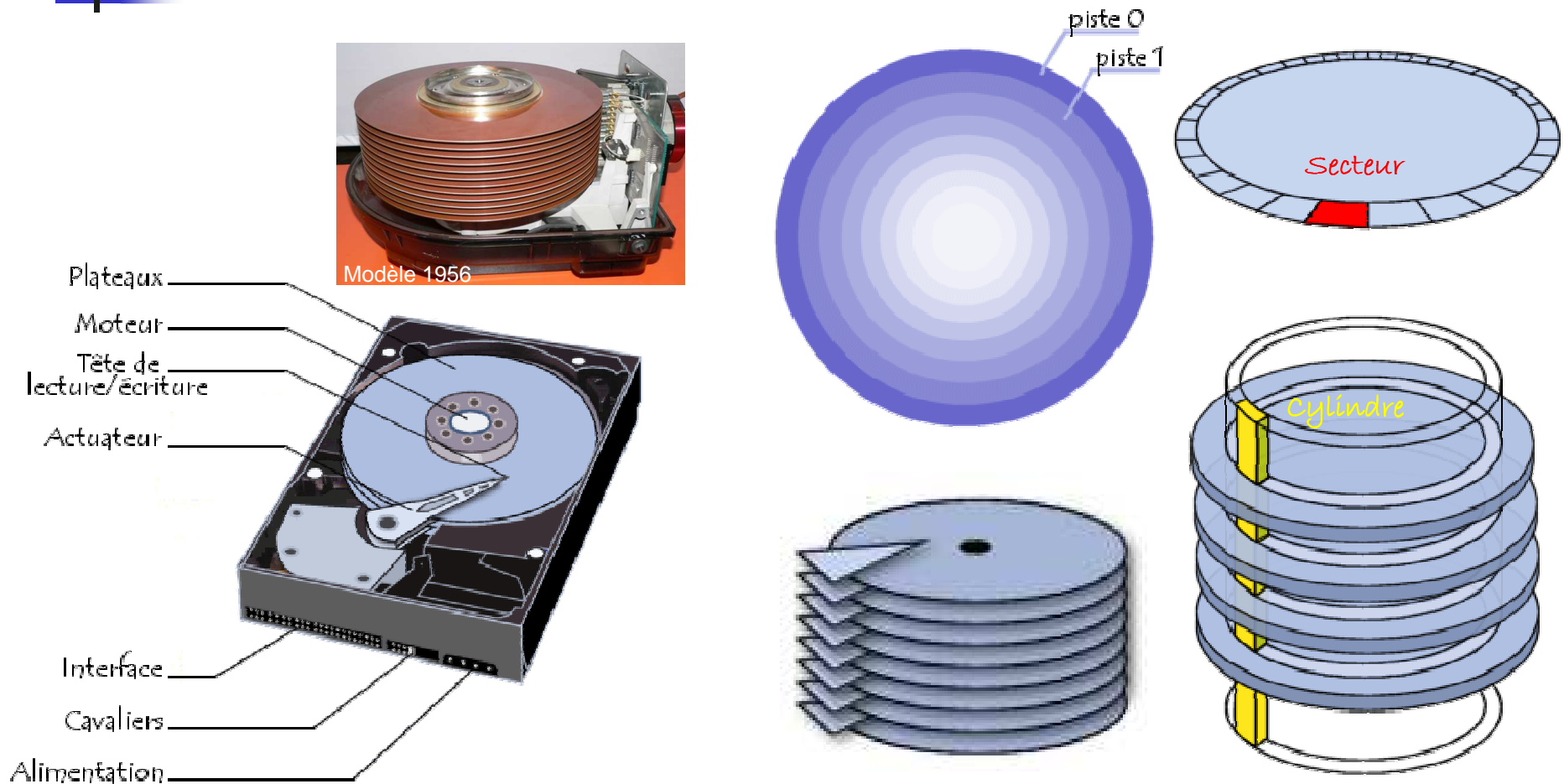
1 MegaOctets =  $2^{20}$  Octets = 1 048 576 Octets

1 GigaOctet =  $2^{30}$  Octets = 1 073 741 824 Octets

1 TeraOctet =  $2^{40}$  Octets = 1 099 511 627 776 O.



## 2. Structuration et codage de l'information Organisation de la mémoire de masse







## 2. Structuration et codage de l'information Organisation de la mémoire de masse

The screenshot shows the 'Défragmenteur de disque' (Disk Defragmenter) window. It displays a table with the following data:

Volume	État de la session	Système de fichiers	Capacité	Espace libre	% Espace libre
(C:)		NTFS	9.77 Go	1.83 Go	18 %
(D:)	Analysé	NTFS	19.53 Go	2.33 Go	11 %
(E:)		NTFS	26.54 Go	26.09 Go	98 %

Below the table, there are two bar charts showing disk space usage before and after defragmentation. The first chart, 'Espace occupé sur le disque avant la défragmentation (estimation)', shows a highly fragmented state with many small red and blue blocks. The second chart, 'Espace occupé sur le disque après la défragmentation (estimation)', shows a more contiguous state with larger blue blocks and less fragmentation. At the bottom, there are buttons for 'Analyser', 'Défragmenter', 'Suspendre', 'Arrêter', and 'Afficher le rapport'. A legend at the bottom indicates: red for 'Fichiers fragmentés', blue for 'Fichiers contigus', green for 'Fichiers non déplaçables', and white for 'Espace libre'.



## 2. Structuration et codage de l'information Organisation de la mémoire de masse

The screenshot shows a Windows 'Rapport d'analyse' window. The title bar reads 'Rapport d'analyse'. The main text states: 'L'analyse est terminée pour : (D:) Il ne vous est pas nécessaire de défragmenter ce volume.' Below this, under 'Informations sur le volume :', a list of statistics is shown for drive (D:):

- Taille du volume = 19,53 Go
- Taille de cluster = 4 Ko
- Espace utilisé = 17,20 Go
- Espace libre = 2,33 Go
- Pourcentage d'espace libre = 11 %
- Fragmentation du volume

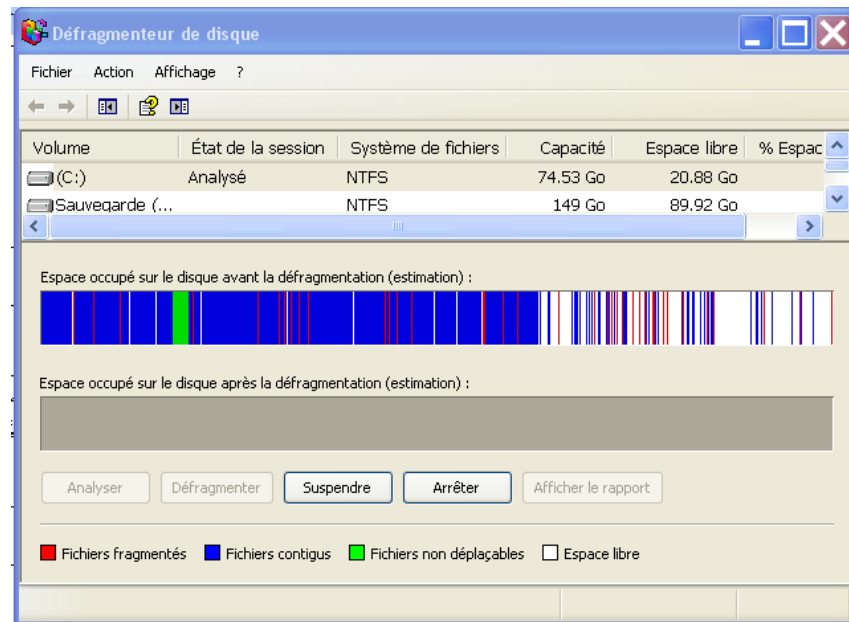
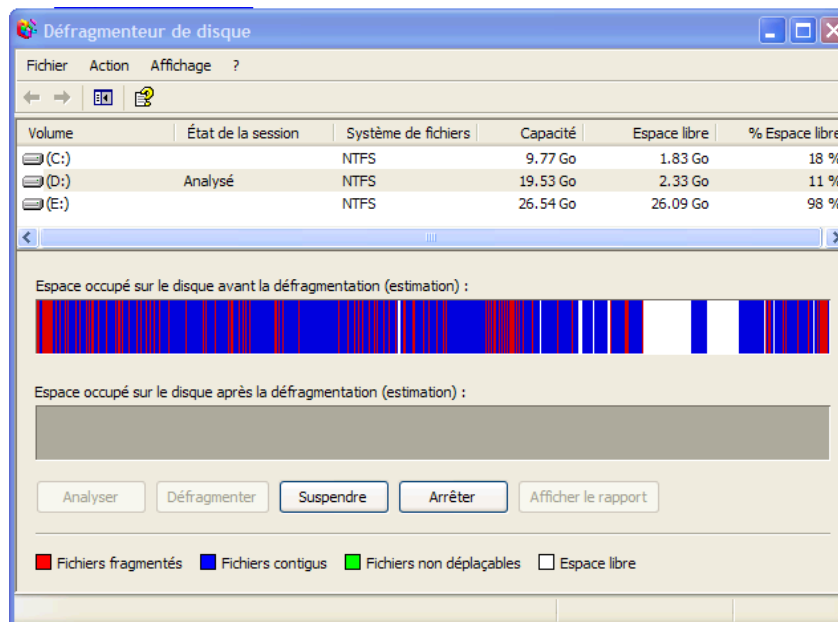
Below the statistics, under 'Fichiers les plus fragmentés:', a table lists the most fragmented files:

Fragments	Taille du fichier	Nom du fichier
238	15 Mo	\\FrancoiseTravail\Enseignement\MasterBDD\S...
212	13 Mo	\\FrancoiseTravail\Enseignement\MasterCCI\C...
205	13 Mo	\\FrancoiseTravail\Enseignement\MasterCCI\C...
200	12 Mo	\\FrancoiseTravail\Enseignement\MasterCCI\C...
185	12 Mo	\\FrancoiseTravail\Enseignement\MasterBDD\S...
179	35 Mo	\\FrancoiseTravail\Administratif\TIMC\CNRS\Pl...
163	10 Mo	\\FrancoiseTravail\Enseignement\MasterCCI\C...
157	10 Mo	\\FrancoiseTravail\Enseignement\MasterBDD\S...
142	9 Mo	\\Romain\SiteGMC\INSAGMC\www-gmc.insa-ly...

At the bottom of the window, there are four buttons: 'Imprimer...', 'Enregistrer sous...', 'Défragmenter', and 'Fermer'.



## 2. Structuration et codage de l'information Organisation de la mémoire de masse





## 2. Structuration et codage de l'information

# Capacité mémoire : Exemple

- une mémoire contenant :  
1024 mots de 32 bits chacun
- la capacité de la mémoire est de  
 $1024 \times 32 = 32768$  bits  
 $32 * \underline{\mathbf{1024}} = 32 \underline{\mathbf{Kilo}}$  bits  
 $1024 * (8*4) = (1024*4) * \underline{\mathbf{8}}$   
 $= (1024*4) \underline{\mathbf{octets}}$   
 $4 * \underline{\mathbf{1024}} * \underline{\mathbf{8}} = 4 \underline{\mathbf{kilo}} \underline{\mathbf{octets}}$  (4 Ko)



## 2. Structuration et codage de l'information

# Supports de sauvegarde et d'archivage

---

- les disquettes : 1.4 Mo et les super-disks : 120 Mo
- le CD Rom R/W : 650 Mo
- les disquettes ZIP : 100 ou 250 Mo
- le DVD R/W : 4.7 Go
- les clefs USB et cartes mémoires : 512 Mo à 64 Go
- la bande magnétique (DAT) jusqu'à 20 Go.
- le disque dur : plusieurs centaines / milliers de Go sur des PC communs



## 2. Structuration et codage de l'information

# Exercice 1

---

Quelle affirmation est correcte ?

1. La RAM et la ROM sont deux types différents de clefs USB.
2. La RAM et la ROM sont deux protocoles de communication différents.
3. La RAM et la ROM sont deux marques/fabricants de bus.
4. La RAM et la ROM sont deux types d'ordinateurs.
5. La RAM et la ROM sont deux types de réseaux.
6. Aucune des affirmations ci-dessus n'est correcte.

Parmi les mots ci-dessous, lequel est la traduction anglaise du mot « octet » ?

1. Bit
2. Byte
3. GIF
4. MPEG
5. Web
6. aucune des réponses ci-dessus



## 2. Structuration et codage de l'information

# Exercice 2

Julien installe sur son site une vidéo de sa dernière compétition. Elle fait 510 000 Ko.

Combien de temps faudra-t-il au minimum pour télécharger ce fichier avec une connexion 512Kbits/s ?

1. entre 1 et 2 secondes
2. entre 5 et 10 secondes
3. entre 1 et 10 minutes
4. entre 30 minutes et 1 heure
5. entre 2 et 3 heures
6. aucune des réponses ci-dessus



## 2. Structuration et codage de l'information

# Codage de l'information

- On associe les suites de bits avec des valeurs.
- Pour un octet, on a
  - $2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = 2^8 = 256$  valeurs
- Avec un octet, on peut représenter 256 informations
- Exemple :
  - codage des nombres
  - table ASCII





## 2. Structuration et codage de l'information

# Un codage

0000 0000 = 0

0000 0001 = 1

0000 0010 = 2

0000 0011 = 3

0000 0100 = 4

0000 0101 = 5

0000 0110 = 6

0000 0111 = 7

0000 1000 = 8

0000 1001 = 9

0000 1010 = 10

0001 0000 = 16

0010 0000 = 32

0100 0000 = 64

1000 0000 = 128

1000 0001 = 129

1000 0010 = 130



## 2. Structuration et codage de l'information

# Des codages

- Codage décimal
  - base 10 ; chiffres 0. 1. 2. 3. 4. 5. 6. 7. 8. 9
  - Ex :  $4037 = 4 \times 10^3 + 0 \times 10^2 + 3 \times 10^1 + 7 \times 10^0$
- Codage binaire
  - base 2 ; chiffres 0 et 1
  - Ex :  $11010 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$   
 $= 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$   
 $= 26$
- Codage hexadécimal
  - base 16 ; chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
  - Ex :  $4f5e = 4 \times 16^3 + 15 \times 16^2 + 5 \times 16^1 + 14 \times 16^0$



## 2. Structuration et codage de l'information

# Des codages : exercice

---

Exprimer 133 en code binaire :

Exprimer 244 en code binaire :



## 2. Structuration et codage de l'information

# Des codages : exercice

Exprimer 133 en code binaire :

128      64      32      16      8      4      2      1

$$133 - 128 = 5$$

$$5 - 4 = 1$$

$$1 - 1 = 0$$

$$1 \times 128 + 0 \times 64 + 0 \times 32 + 0 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\ = 133 = 10000101$$



## 2. Structuration et codage de l'information

# Des codages : exercice

Exprimer 244 en code binaire :

128      64      32      16      8      4      2      1

$$244 - 128 = 116$$

$$116 - 64 = 52$$

$$52 - 32 = 20$$

$$20 - 16 = 4$$

$$4 - 4 = 0$$

$$1 \times 128 + 1 \times 64 + 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$$

$$= 244 = 11110100$$



## 2. Structuration et codage de l'information

### Exemple 10100011 : de binaire à héra

- Solution 1

$$= 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$= [163]_{10}$$

$$= 16 \cdot 10 + 3 = A \cdot 16 + 3 = A3$$

- Solution 2

4 bits permettent de coder 16 caractères

Décomposer les octets en 2 paquets de 4 bits

Les traduire séparément, puis rassembler

$$\left. \begin{array}{l} [1010]_2 = [10]_{10} = [A]_{16} \\ [0011]_2 = [3]_{10} = [3]_{16} \end{array} \right\} A3$$



## 2. Structuration et codage de l'information

# De l'hexadécimal au binaire

---

- Comment s'écrit [FF]16 en binaire ?



## 2. Structuration et codage de l'information

# Codage ASCII

---

- On peut associer différentes significations à une suite de caractères
- Table ASCII
  - Association valeur numérique / caractère
  - Utilisé pour traduire des textes (écrits de façon alphabétique) en suite d'octet.





## 2. Structuration et codage de l'information

# Codage ASCII

- 7bits -> 128 caractères
- 26 lettres majuscules A – Z
- 26 lettres minuscule a - z
- 10 chiffres 0 à 9
- 33 caractères de ponctuation :  
sp, ! " # \$ % & ' ( ) \* + , - . / < = > ? @ [ ] ^ \_ ` { | } ~
- 33 caractères de contrôle :  
null, etx, bel, bs, ht, lf, vt, ff, cr, ..., del



## 2. Structuration et codage de l'information

# Codage ASCII

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

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



## 2. Structuration et codage de l'information

# Codage ASCII

Le code binaire ci-dessous représente un mot de 7 lettres ASCII. Décodez-le.

```
01000010  
01001001  
01001110  
01000001  
01001001  
01010010  
01000101
```



## 2. Structuration et codage de l'information

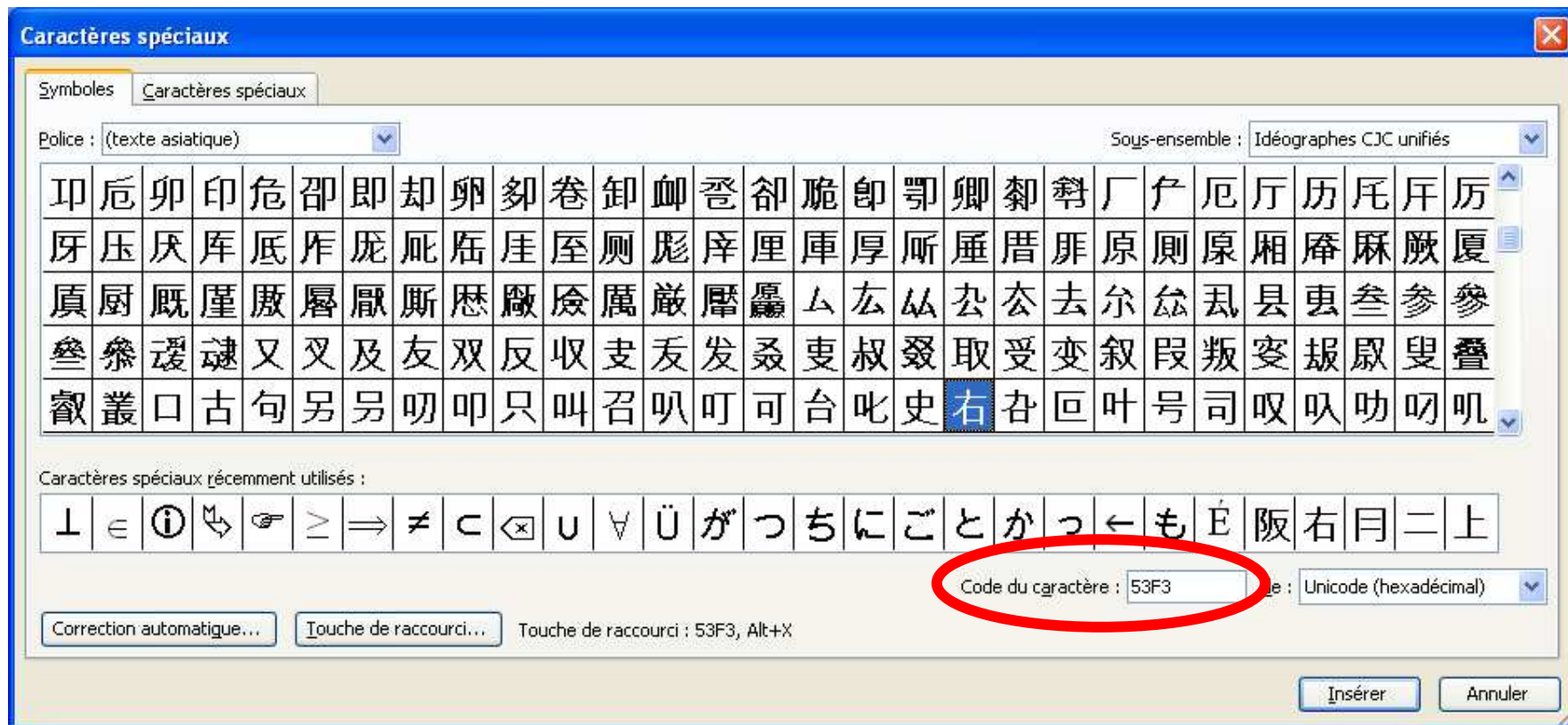
# Codage ASCII étendu

- Avec un octet, on code 256 caractères
- Les codes ISO 8859-1 à 8859-11 définissent les caractères entre 128 et 255
- Il existe plus de
  - 4000 caractères chinois (Kanjis)
  - Différents alphabets
- Il existe d'autres codages (étendus)
  - Sur 2 ou 3 octets.
  - Chaque caractère = 2 ou 3 octets
  - UNICODE 16 bits -> 65 536 caractères
  - UCS-2 16 bits - UCS-4 32 bits



## 2. Structuration et codage de l'information

# Codage UNICODE





# Corrections Exercices

---



# Exercice : décodage binaire

---

Le code binaire ci-dessous représente un mot de 7 lettres ASCII. Décodez-le.

```
01000010  
01001001  
01001110  
01000001  
01001001  
01010010  
01000101
```



## décodage binaire solution par la traduction en hexadécimal

---

Le code binaire ci-dessous représente un mot de 7 lettres ASCII. Décodez-le.

<b>0100</b>	<b>0010</b>	4 2
<b>0100</b>	<b>1001</b>	4 9
<b>0100</b>	<b>1110</b>	4 [14] = 4 E
<b>0100</b>	<b>0001</b>	4 1
<b>0100</b>	<b>1001</b>	4 9
<b>0101</b>	<b>0010</b>	5 2
<b>0100</b>	<b>0101</b>	4 5





# décodage binaire solution par la traduction en hexadécimal

Le code binaire ci-dessous représente un mot de 7 lettres ASCII. Décodez-le.

<b>0100</b>	<b>0010</b>	4 2	<b>B</b>
<b>0100</b>	<b>1001</b>	4 9	<b>I</b>
<b>0100</b>	<b>1110</b>	4 [14] = 4 E	<b>N</b>
<b>0100</b>	<b>0001</b>	4 1	<b>A</b>
<b>0100</b>	<b>1001</b>	4 9	<b>I</b>
<b>0101</b>	<b>0010</b>	5 2	<b>R</b>
<b>0100</b>	<b>0101</b>	4 5	<b>E</b>



# décodage binaire

## solution par la traduction en décimal

---

Le code binaire ci-dessous représente un mot de 7 lettres ASCII. Décodez-le.

0100	0010	$1x64 + 0x32 + 0x16 + 0x8 + 0x4 + 1x2 + 0x1 = 66$
0100	1001	$1x64 + 0x32 + 0x16 + 1x8 + 0x4 + 0x2 + 1x1 = 73$
0100	1110	$1x64 + 0x32 + 0x16 + 1x8 + 1x4 + 1x2 + 0x1 = 78$
0100	0001	$1x64 + 0x32 + 0x16 + 0x8 + 0x4 + 0x2 + 1x1 = 65$
0100	1001	$1x64 + 0x32 + 0x16 + 1x8 + 0x4 + 0x2 + 1x1 = 73$
0101	0010	$1x64 + 0x32 + 1x16 + 0x8 + 0x4 + 1x2 + 0x1 = 82$
0100	0101	$1x64 + 0x32 + 0x16 + 0x8 + 1x4 + 0x2 + 1x1 = 69$



# décodage binaire

## solution par la traduction en décimal

---

Le code binaire ci-dessous représente un mot de 7 lettres ASCII. Décodez-le.

01000010	= 66	B
01001001	= 73	I
01001110	= 78	N
01000001	= 65	A
01001001	= 73	I
01010010	= 82	R
01000101	= 69	E



# décodage binaire

## Décodage avec Excel

	A	B	C	D	E	F	G
1			Décimal	Binaire	hexadécimal	Octal	codeASCII
63	61	61	61	00111101	3D	075	=
64	62	62	62	00111110	3E	076	>
65	63	63	63	00111111	3F	077	?
66	64	64	64	01000000	40	100	@
67	65	65	65	01000001	41	101	A
68	66	66	66	01000010	42	102	B
69	67	67	67	01000011	43	103	C

	A	B	C	D	E	F	G
1			Décimal	Binaire	hexadécimal	Octal	codeASCII
63	=A62+1	61	61	=DECBIN(C63;8)	=DECHEX(C63;2)	=DECOCT(C63;3)	=CAR(C63)
64	=A63+1	62	62	=DECBIN(C64;8)	=DECHEX(C64;2)	=DECOCT(C64;3)	=CAR(C64)
65	=A64+1	63	63	=DECBIN(C65;8)	=DECHEX(C65;2)	=DECOCT(C65;3)	=CAR(C65)
66	=A65+1	64	64	=DECBIN(C66;8)	=DECHEX(C66;2)	=DECOCT(C66;3)	=CAR(C66)
67	=A66+1	65	65	=DECBIN(C67;8)	=DECHEX(C67;2)	=DECOCT(C67;3)	=CAR(C67)
68	=A67+1	66	66	=DECBIN(C68;8)	=DECHEX(C68;2)	=DECOCT(C68;3)	=CAR(C68)
69	=A68+1	67	67	=DECBIN(C69;8)	=DECHEX(C69;2)	=DECOCT(C69;3)	=CAR(C69)
70	=A69+1	68	68	=DECBIN(C70;8)	=DECHEX(C70;2)	=DECOCT(C70;3)	=CAR(C70)

Menu *Outils* => Macros complémentaires ...

=> Utilitaires d'analyse

Menu *Insertion* => fonctions (fonctions scientifiques)