

# Rien que des 0 et 1 : des bits!



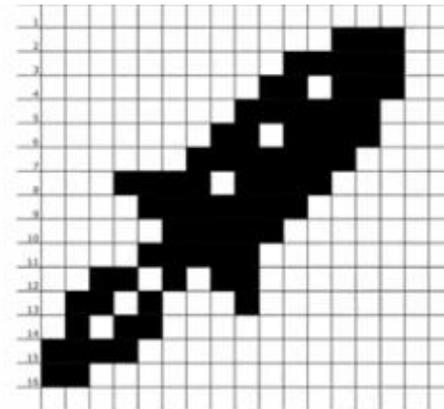
**Acknowledgment and Disclaimer:** La version anglaise de cette présentation est issue de cours mobile CSP de Ralph Morelli et al.. Elle a été en partie financée par la National Science Foundation (réf 1240841, 1225680, 1225719, 1225745, 1225976, and 1226216).

Traduction et adaptation françaises réalisées par <https://onvaessayer.org> Pierre Huguete

# L'encodage run Length (RLE)

- Compression RLE
- # pixels blancs, # pixels noirs, # blancs, ...
  - 16: la ligne 1 a 16 pixels blancs
  - 12, 3, 1 : la ligne 2, 12 blancs, 3 noirs, 1 blanc

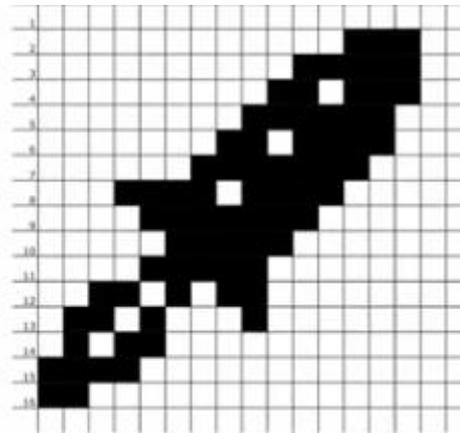
```
16
12, 3, 1
10, 5, 1
9, 2, 1, 3, 1
8, 6, 2
7, 2, 1, 4, 2
6, 7, 3
3, 4, 1, 4, 4
4, 7, 5
5, 5, 6
4, 5, 7
2, 2, 1, 1, 1, 2, 7
1, 2, 1, 1, 3, 1, 7
1, 1, 1, 2, 11
0, 4, 12
0, 2, 14
```



# L'encodage run Length (RLE)

- RLE est une méthode de compression *sans perte*.
- adaptée à des images avec beaucoup de blanc
  - par exemple, les Fax
- Utilisée dans le format BMP (*bitmap*)

```
16
12, 3, 1
10, 5, 1
9, 2, 1, 3, 1
8, 6, 2
7, 2, 1, 4, 2
6, 7, 3
3, 4, 1, 4, 4
4, 7, 5
5, 5, 6
4, 5, 7
2, 2, 1, 1, 1, 2, 7
1, 2, 1, 1, 3, 1, 7
1, 1, 1, 2, 11
0, 4, 12
0, 2, 14
```



# Compressions avec ou sans perte

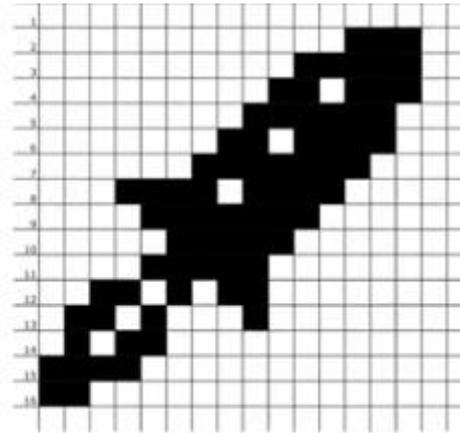
- **Sans perte:** l'image originale peut être reconstruite à l'identique à partir des données compressées
  - adaptée aux images médicales, ...
  - utilisée dans le format d'image BMP (**bitmap**)
- **Avec perte:** des données peuvent être perdues par le processus de compression/décompression
  - Ok pour les photos où l'oeil humain ne peut pas voir la différence,
  - par exemple, le format JPEG

# Volume des données?

RLE

Original Image

```
16  
12, 3, 1  
10, 5, 1  
9, 2, 1, 3, 1  
8, 6, 2  
7, 2, 1, 4, 2  
6, 7, 3  
3, 4, 1, 4, 4  
4, 7, 5  
5, 5, 6  
4, 5, 7  
2, 2, 1, 1, 1, 2, 7  
1, 2, 1, 1, 3, 1, 7  
1, 1, 1, 2, 11  
0, 4, 12  
0, 2, 14
```



62 nombres

256 pixels

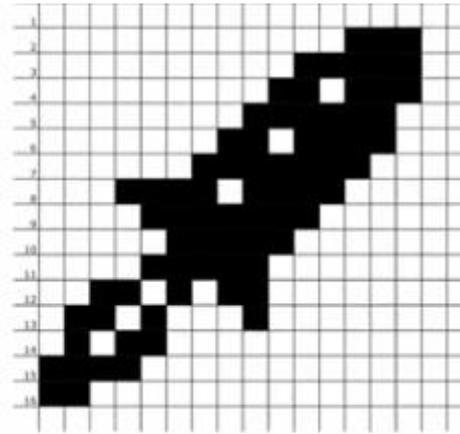
16 x 16

# Taux de compression?

RLE

Original Image

```
16  
12, 3, 1  
10, 5, 1  
9, 2, 1, 3, 1  
8, 6, 2  
7, 2, 1, 4, 2  
6, 7, 3  
3, 4, 1, 4, 4  
4, 7, 5  
5, 5, 6  
4, 5, 7  
2, 2, 1, 1, 1, 2, 7  
1, 2, 1, 1, 3, 1, 7  
1, 1, 1, 2, 11  
0, 4, 12  
0, 2, 14
```



62 nombres

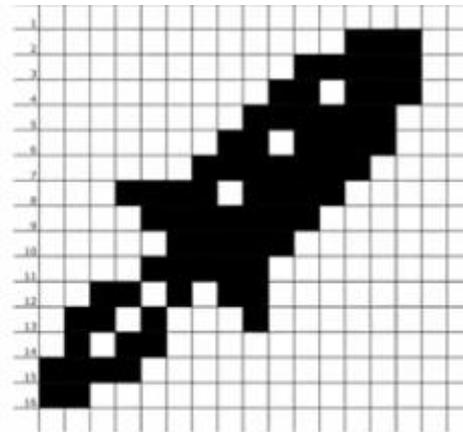
256 pixels

16 x 16

compression? Dépend du nombre de *bits par pixel*

# Image Monochrome

- 1 bit par pixel
- 256 pixels => 256 bits
- 1 = blanc, 0 = noir
- Technique des années 1960...



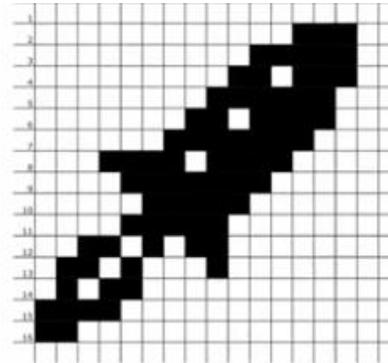
```
11111111111111111111
11111111111110001
11111111111000001
11111111110010001
1111111100000011
1111111001000011
1111110000000111
1110000100001111
1111000000111111
1111100000111111
1111000001111111
1100101001111111
1001011101111111
1010011111111111
0000111111111111
0011111111111111
```

# Monochrome: volume de données?

Image  
compressée

```
16  
12, 3, 1  
10, 5, 1  
9, 2, 1, 3, 1  
8, 6, 2  
7, 2, 1, 4, 2  
6, 7, 3  
3, 4, 1, 4, 4  
4, 7, 5  
5, 5, 6  
4, 5, 7  
2, 2, 1, 1, 1, 2, 7  
1, 2, 1, 1, 3, 1, 7  
1, 1, 1, 2, 11  
0, 4, 12  
0, 2, 14
```

Image  
originale

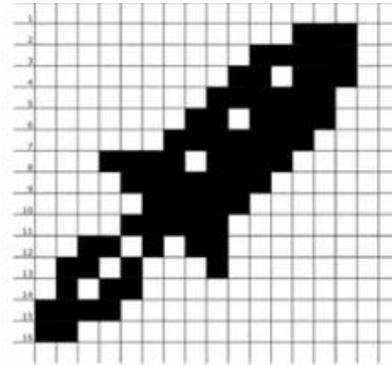


# Monochrome: How much data?

Compressed  
Image

```
16  
12, 3, 1  
10, 5, 1  
9, 2, 1, 3, 1  
8, 6, 2  
7, 2, 1, 4, 2  
6, 7, 3  
3, 4, 1, 4, 4  
4, 7, 5  
5, 5, 6  
4, 5, 7  
2, 2, 1, 1, 1, 2, 7  
1, 2, 1, 1, 3, 1, 7  
1, 1, 1, 2, 11  
0, 4, 12  
0, 2, 14
```

Original  
Image



256 pixels  
x 1 bit/pixel  
256 bits

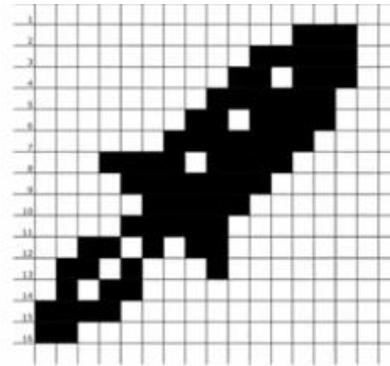
# Monochrome: How much data?

Compressed  
Image

```
16  
12, 3, 1  
10, 5, 1  
9, 2, 1, 3, 1  
8, 6, 2  
7, 2, 1, 4, 2  
6, 7, 3  
3, 4, 1, 4, 4  
4, 7, 5  
5, 5, 6  
4, 5, 7  
2, 2, 1, 1, 1, 2, 7  
1, 2, 1, 1, 3, 1, 7  
1, 1, 1, 2, 11  
0, 4, 12  
0, 2, 14
```

62 nombres  
x 8 bit/nombre  
496 bits

Original  
Image



256 pixels  
x 1 bit/pixel  
256 bits

# Monochrome: How much data?

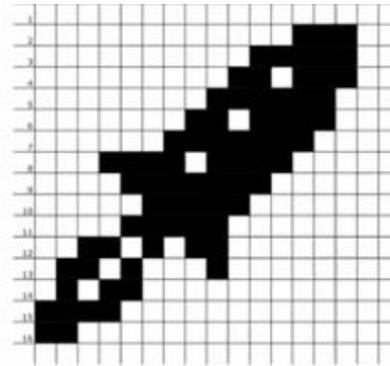
Compressed  
Image

```
16
12, 3, 1
10, 5, 1
9, 2, 1, 3, 1
8, 6, 2
7, 2, 1, 4, 2
6, 7, 3
3, 4, 1, 4, 4
4, 7, 5
5, 5, 6
4, 5, 7
2, 2, 1, 1, 1, 2, 7
1, 2, 1, 1, 3, 1, 7
1, 1, 1, 2, 11
0, 4, 12
0, 2, 14
```

62 nombres  
x 8 bit/nombre

496 bits

Original  
Image



256 pixels  
x 1 bit/pixel

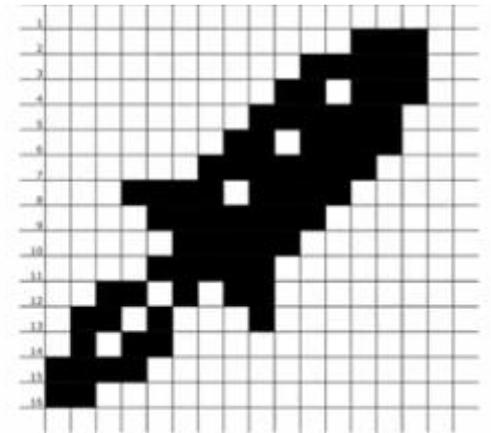
256 bits



# Image en 256 couleurs : 8 bit/pixel

- Technique des années 1980...
- Rouge, Vert, Bleu (RVB)
- 8 bits par pixel
- 3 R, 3 V, 2 B

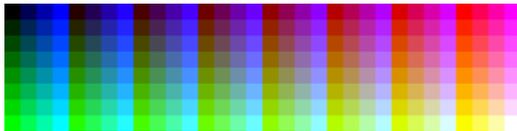
0	1	2	3	4	5	6	7
R	R	R	V	V	V	B	B
1	1	1	0	0	0	1	1



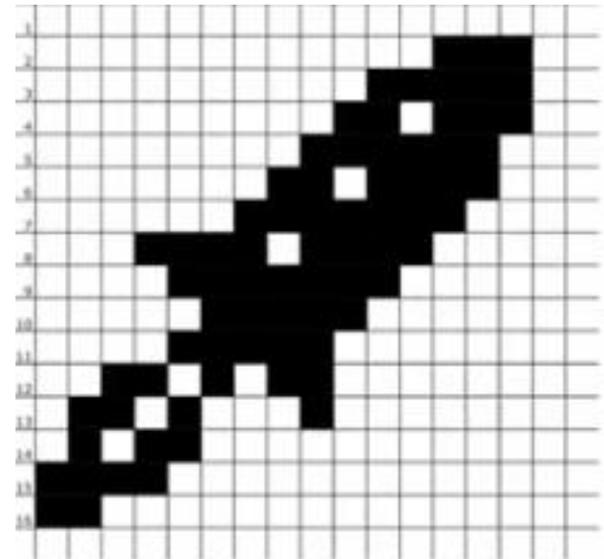
- 0000 0000 = noir(0)
- 1111 1111 = blanc (255)

# RLE modifié

Si il y a plus de 2 couleurs, il faut 2 nombres par plage : pour la couleur et la longueur de la plage



```
                255,16
                255,12, 0,3 255,1
                255,10, 0,5, 255,1
                255,9, 0,2, 255,1, 0,3, 255,1
                255,8, 0,6, 255,2
                255,7, 0,2, 255,1, 0,4, 255,2
                255,6, 0,7, 255,3
                255,3, 0,5, 255,1, 0,4, 255,4
                255,4, 0,7, 255,5
                255,5, 0,5, 255,6
                255,4, 0,5, 255,7
                255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7
                255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7
                255,1, 0,1, 255,1, 0, 2, 255, 11
                255,0, 0,4, 255,12
                255,0, 0,2, 255,14
```

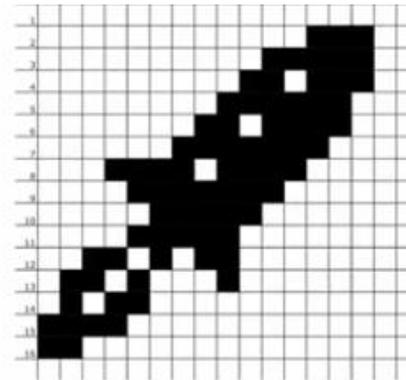


# RLE modifié

Compressed  
Image

Original  
Image

```
                255,16
                255,12, 0,3 255,1
                255,10, 0,5, 255,1
255,9, 0,2, 255,1, 0,3, 255,1
                255,8, 0,6, 255,2
255,7, 0,2, 255,1, 0,4, 255,2
                255,6, 0,7, 255,3
255,3, 0,5, 255,1, 0,4, 255,4
                255,4, 0,7, 255,5
                255,5, 0,5, 255,6
                255,4, 0,5, 255,7
255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7
255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7
                255,1, 0,1, 255,1, 0, 2, 255, 11
                255,0, 0,4, 255,12
                255,0, 0,2, 255,14
```

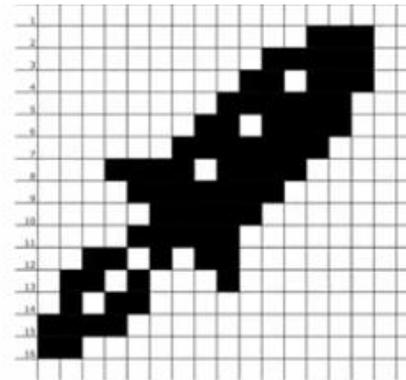


# RLE modifié

Image  
compressée

Image  
originale

255,16  
255,12, 0,3 255,1  
255,10, 0,5, 255,1  
255,9, 0,2, 255,1, 0,3, 255,1  
255,8, 0,6, 255,2  
255,7, 0,2, 255,1, 0,4, 255,2  
255,6, 0,7, 255,3  
255,3, 0,5, 255,1, 0,4, 255,4  
255,4, 0,7, 255,5  
255,5, 0,5, 255,6  
255,4, 0,5, 255,7  
255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7  
255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7  
255,1, 0,1, 255,1, 0, 2, 255, 11  
255,0, 0,4, 255,12  
255,0, 0,2, 255,14



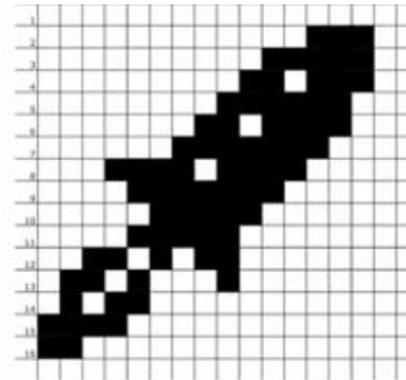
256 pixels  
x 8 bits/pixel  
2048 bits

# RLE modifié

Image  
compressée

Image  
originale

255,16  
255,12, 0,3 255,1  
255,10, 0,5, 255,1  
255,9, 0,2, 255,1, 0,3, 255,1  
255,8, 0,6, 255,2  
255,7, 0,2, 255,1, 0,4, 255,2  
255,6, 0,7, 255,3  
255,3, 0,5, 255,1, 0,4, 255,4  
255,4, 0,7, 255,5  
255,5, 0,5, 255,6  
255,4, 0,5, 255,7  
255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7  
255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7  
255,1, 0,1, 255,1, 0, 2, 255, 11  
255,0, 0,4, 255,12  
255,0, 0,2, 255,14



124 nombres  
x 8 bits/nombre  
992 bits

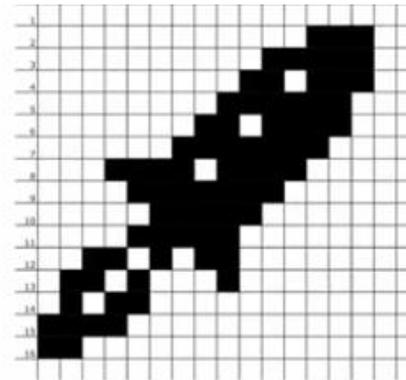
256 pixels  
x 8 bits/pixel  
2048 bits

# RLE modifié

Image  
compressée

Image  
originale

```
255,16
255,12, 0,3 255,1
255,10, 0,5, 255,1
255,9, 0,2, 255,1, 0,3, 255,1
255,8, 0,6, 255,2
255,7, 0,2, 255,1, 0,4, 255,2
255,6, 0,7, 255,3
255,3, 0,5, 255,1, 0,4, 255,4
255,4, 0,7, 255,5
255,5, 0,5, 255,6
255,4, 0,5, 255,7
255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7
255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7
255,1, 0,1, 255,1, 0, 2, 255, 11
255,0, 0,4, 255,12
255,0, 0,2, 255,14
```



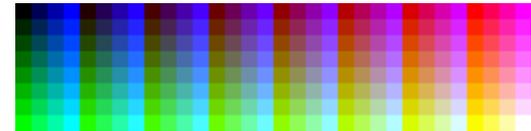
124 nombres  
x 8 bits/nombre  
**992 bits**

256 pixels  
x 8 bits/pixel  
**2048 bits**

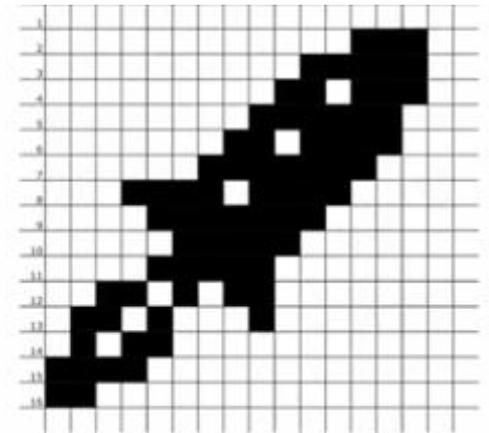


# 16 millions de couleurs sur 24 bits

- Red, Vert, Bleu (RVB)
- 24 bits par pixel
  - 8 R, 8 V, 8 B
- $2^{24} = 16,777,216$  couleurs
- Plus que l'on peut voir



- 3 nombres par pixel
- 0,0,0 = noir
- 255,255,255 = blanc
- 255, 0, 200 = 



# 16 millions de couleurs sur 24 bits

4 fois plus de nombres

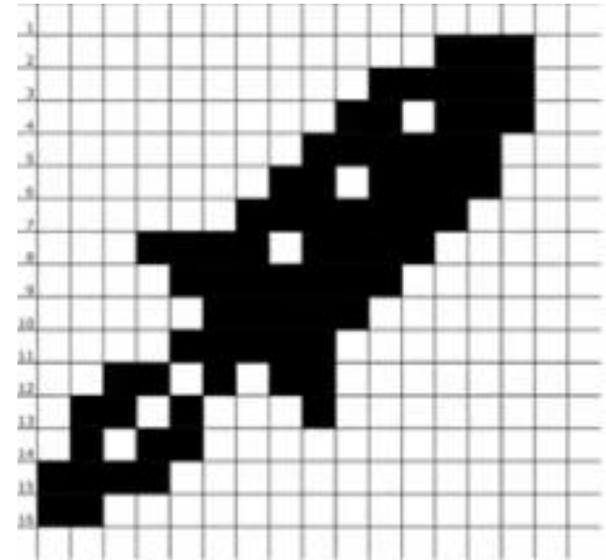
248 nombres avec 8 bits/nombre

1984 bits

255,255,255,16  
255,255,255,12, 0,0,0,3 255,255,255,1

...

255,255,255,0, 0,0,0,4, 255,255,255,12  
255,255,255,0, 0,0,0,2, 255,255,255,14



# 16 millions de couleurs sur 24 bits

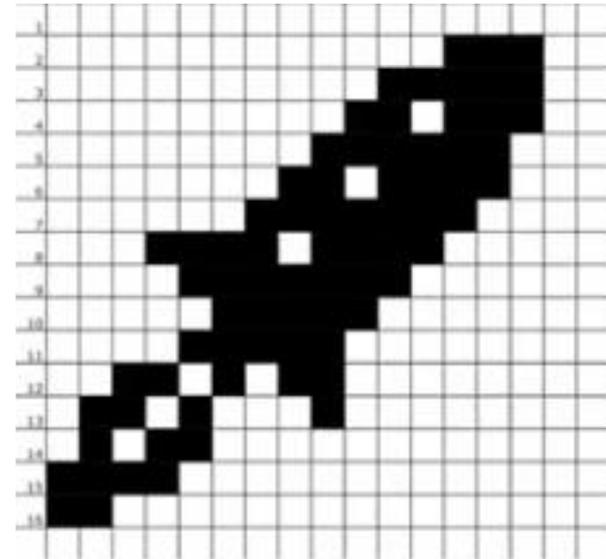
Compressed  
Image

255,255,255,16  
255,255,255,12, 0,0,0,3 255,255,255,1

...

255,255,255,0, 0,0,0,4, 255,255,255,12  
255,255,255,0, 0,0,0,2, 255,255,255,14

Original  
Image



# 16 millions de couleurs sur 24 bits

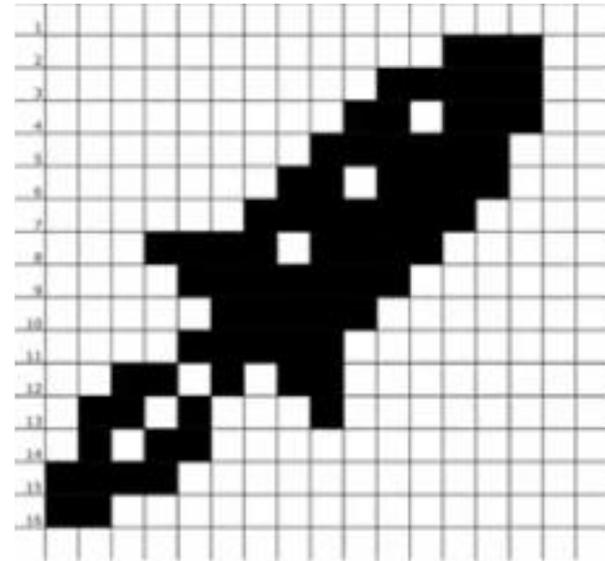
Image  
compressée

255,255,255,16  
255,255,255,12, 0,0,0,3 255,255,255,1

...

255,255,255,0, 0,0,0,4, 255,255,255,12  
255,255,255,0, 0,0,0,2, 255,255,255,14

Image  
originale



256 pixels  
x 24 bits/pixel  
6,144 bits

# 16 millions de couleurs sur 24 bits

Image  
compressée

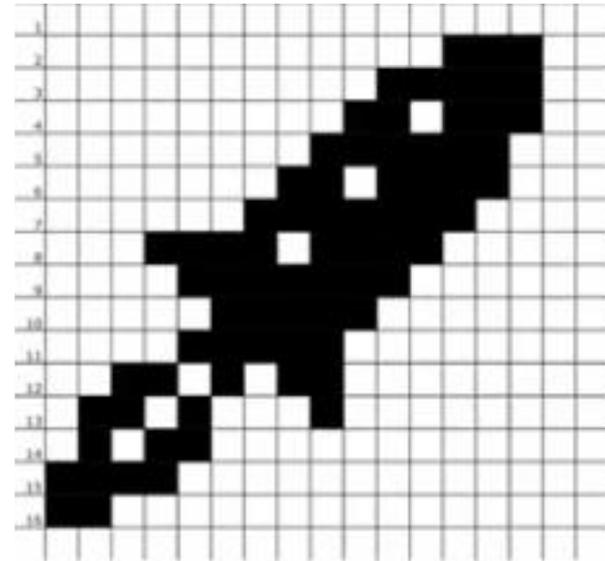
255,255,255,16  
255,255,255,12, 0,0,0,3 255,255,255,1

...

255,255,255,0, 0,0,0,4, 255,255,255,12  
255,255,255,0, 0,0,0,2, 255,255,255,14

248 nombres  
x 8 bits/nombre  
1984 bits

Image  
originale



256 pixels  
x 24 bits/pixel  
6,144 bits

# 16 millions de couleurs sur 24 bits

Image  
compressée

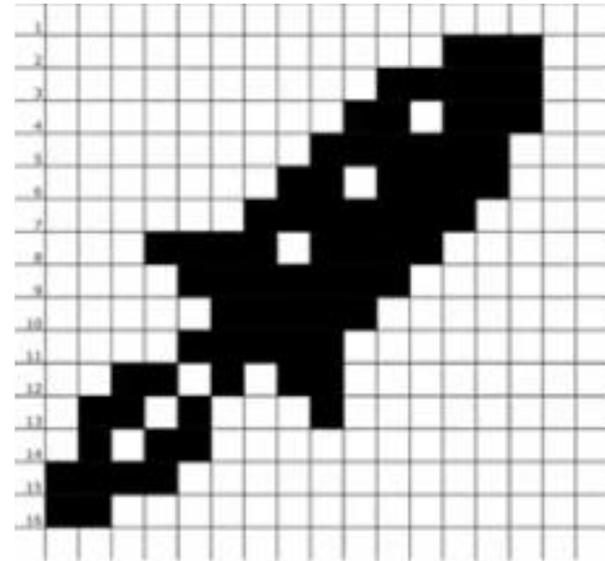
255,255,255,16  
255,255,255,12, 0,0,0,3 255,255,255,1

...

255,255,255,0, 0,0,0,4, 255,255,255,12  
255,255,255,0, 0,0,0,2, 255,255,255,14

248 nombres  
x 8 bits/nombre  
**1984 bits**

Image  
originale

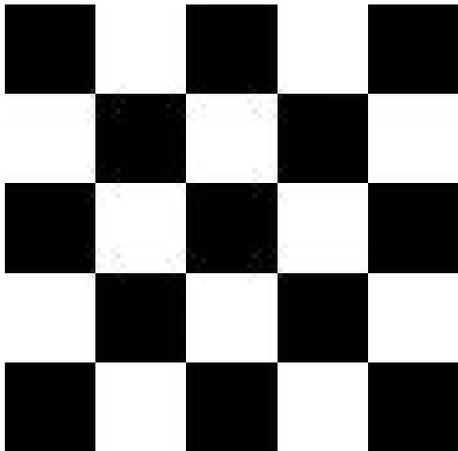


256 pixels  
x 24 bits/pixel  
**6,144 bits**

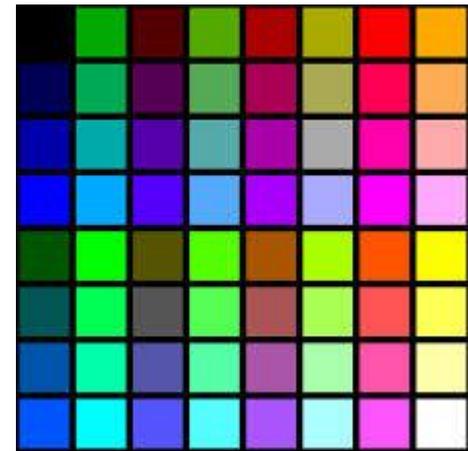


# Quel taux de compression RLE?

- Dépends du nombre de bits/pixel
- Mais, dépend encore plus...
- ...*du nombre de couleurs différentes dans l'image*



Plus il y a de couleurs, plus elles changent, donc plus il faut de nombres.



Nous avons vu que le système binaire permettrait de

- représenter des nombres
- représenter des couleurs
- représenter un langage machine

# Représentation des caractères

- Le codage ASCII des caractères (1960...)  
American Standard Code for Information
- Codage sur 7-bit : 128 caractères

ASCII table

Hex	Symbol										
20		30	0	40	@	50	P	60	`	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	\$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	

# Représentation des caractères

- ASCII : quelques exemples

Binary	Hex	Decimal	Character
0010 0000	20	32	Space
0011 0000	30	48	0
0100 0001	41	65	A
0110 0001	61	97	a

Donc,...

Quel est le caractère correspondant à  
cette séquence de bits ?

0100 0001

Est-ce que c'est le nombre 65?  
ou la lettre 'A'  
ou la couleur 

Donc,...

Quel est le caractère correspondant à  
cette séquence de bits ?

0100 0001

Donc,...

Quel est le caractère correspondant à  
cette séquence de bits ?

0100 0001

Est-ce que c'est le nombre 65 ?

Donc,...

Quel est le caractère correspondant à  
cette séquence de bits ?

0100 0001

Est-ce que c'est le nombre 65 ?  
ou la lettre 'A' ?

Donc,...

Quel est le caractère correspondant à  
cette séquence de bits ?

0100 0001

Est-ce que c'est le nombre 65 ?

ou la lettre 'A'

ou la couleur 

Ca dépend!

si

0100 0001

Est utilisé pour

une Image ?

un Texte ?

un nombre ?

ou le langage machine ?

Rien que des 0 et 1 :  
des bits!