

Série 2: Représentation de textes et d'images

Exercice 1 (Coder un texte en binaire, avec corrigé détaillé)

Trouver la représentation binaire en ASCII du texte « Je pense, donc je suis. »

Exercice 2 (Coder un texte en binaire)

Trouver la représentation binaire en ASCII du texte "Cet exercice est un peu fastidieux."

Exercice 3

Écrire les nombres suivants en base hexadécimal :

1. $10010010110_{(2)}$
2. $111110_{(2)}$
3. $1000110101110101_{(2)}$
4. $11110000000011_{(2)}$

Exercice 4 (Lire un texte écrit en binaire, avec corrigé détaillé)

Trouver le texte représenté en ASCII binaire par la suite de bits

01000011 00100111 01100101 01110011 01110100 00100000 01100110 01100001 01100011
01101001 01101100 01100101.

Exercice 5 (Lire un texte écrit en binaire)

Trouver le texte représenté en ASCII binaire par la suite de bits

00110000 01110100 01100101 01110100 01110100 00110001.

Exercice 6 (Éditeur hexadécimal)

1. Ouvrir l'éditeur de texte "kate", puis créer un fichier texte qui contient le texte « Je pense, donc je suis. ». Sauver le fichier texte sous le nom "ex_phrase.txt".
2. Ouvrir le fichier avec un éditeur hexadécimal, par exemple l'éditeur en ligne HexEd (<https://hexed.it/>) ou "Okteta" (si installé).
3. Donner la représentation en hexadécimal du code ASCII du fichier.

Exercice 7 (Nécessité de méta-données)

Une image a été codée avec la suite de bits suivante

0000000000001111100011000011001000000100100000010010000001001000000100110000110
00111111000000000000

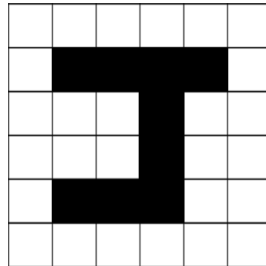
1. Pouvez-vous déterminer ce que représente cette image ?
2. Même question mais sachant que cette suite de bits représente une image de dimension 10 x 10.

Exercice 8 (Codage binaire d'image)

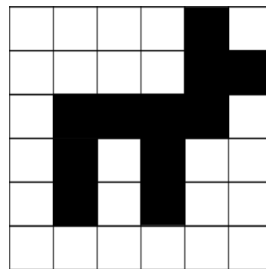
Coder chaque image en utilisant 1 bit par pixel (même codage qu'étudié dans la théorie). Indiquer les dimensions de chaque image

Binaire

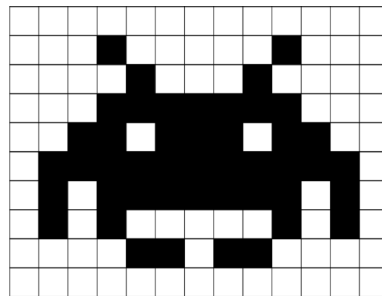
a)



b)



c)



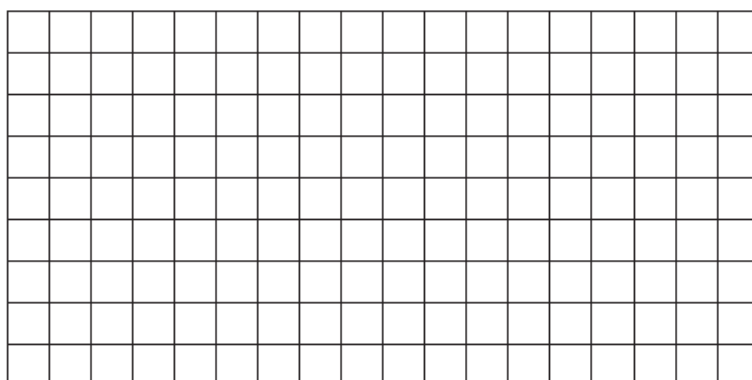
Exercice 9 (Un autre codage binaire d'image)

Un nouvelle représentation d'images en noir et blanc est proposé. Dans ce système chaque ligne est représentée par un suite de nombre qui code l'alternance entre pixel blanc et noir.

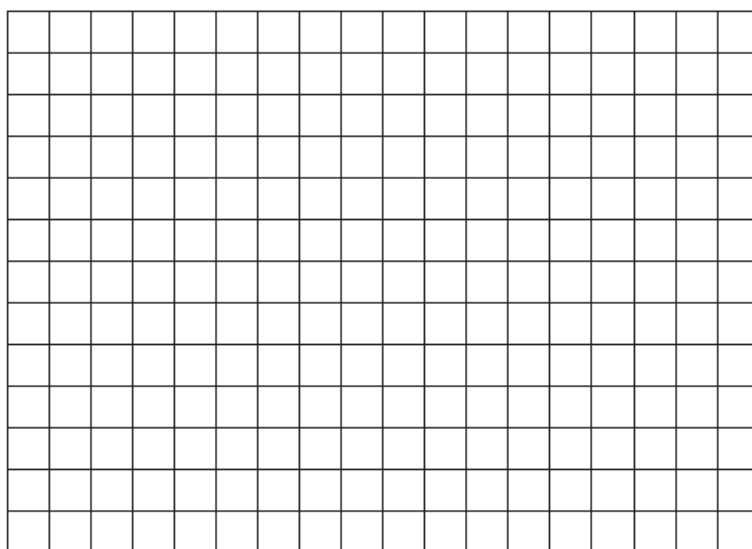
						1, 3, 1
						4, 1
						1, 4
						0, 1, 3, 1
						0, 1, 3, 1
						1, 4

L'image ci-dessus nous montre illustre le codage proposé. La première ligne contient un pixel blanc, trois noirs puis un blanc. Ainsi, la première ligne est représentée par 1, 3, 1. Le premier nombre représente toujours le nombre de pixels blancs. Si le premier pixel est noir, la ligne commencera par un 0.

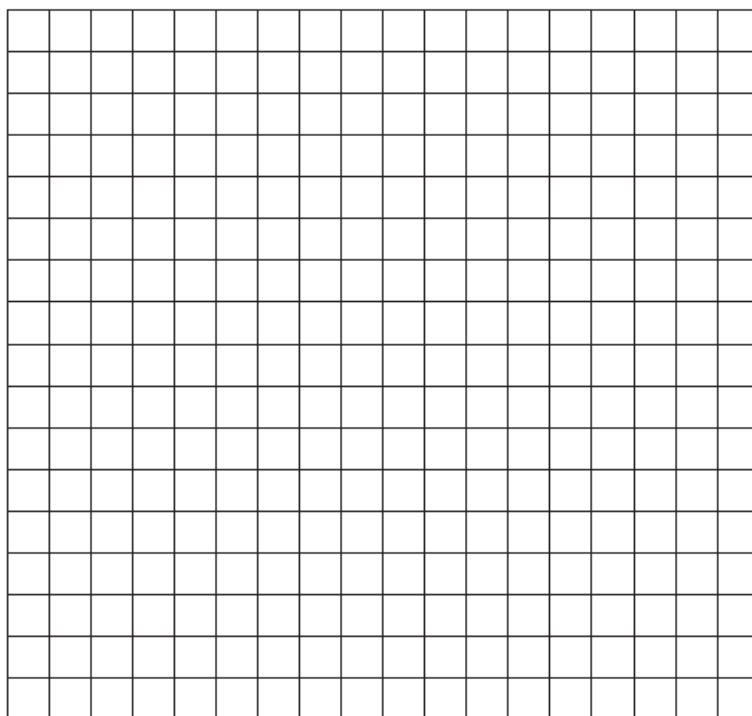
Décoder les image ci-dessous en comptant les grilles de pixels.



4, 11
4, 9, 2, 1
4, 9, 2, 1
4, 11
4, 9
4, 9
5, 7
0, 17
1, 15



6, 5, 2, 3
4, 2, 5, 2, 3, 1
3, 1, 9, 1, 2, 1
3, 1, 9, 1, 1, 1
2, 1, 11, 1
2, 1, 10, 2
2, 1, 9, 1, 1, 1
2, 1, 8, 1, 2, 1
2, 1, 7, 1, 3, 1
1, 1, 1, 1, 4, 2, 3, 1
0, 1, 2, 1, 2, 2, 5, 1
0, 1, 3, 2, 5, 2
1, 3, 2, 5



6, 2, 2, 2
5, 1, 2, 2, 2, 1
6, 6
4, 2, 6, 2
3, 1, 10, 1
2, 1, 12, 1
2, 1, 3, 1, 4, 1, 3, 1
1, 2, 12, 2
0, 1, 16, 1
0, 1, 6, 1, 2, 1, 6, 1
0, 1, 7, 2, 7, 1
1, 1, 14, 1
2, 1, 12, 1
2, 1, 5, 2, 5, 1
3, 1, 10, 1
4, 2, 6, 2
6, 6

Corrigé Série 2 :

Ex 1

On cherche la table des codes ASCII sur le Web de manière à traduire le texte, caractère par caractère : 74, 101, 32, 112, 101, 110, 115, 101, 44, 32, 100, 111, 110, 99, 32, 106, 101, 32, 115, 117, 105, 115, 46. On exprime ensuite chacun de ces nombres en binaire sur huit bits :
 01001010 01100101 00100000 01110000 01100101 01101110 01110011 01100101 00101100
 00100000 01100100 01101111 01101110 01100011 00100000 01101010 01100101 00100000
 01110011 01110101 01101001 01110011 00101110.

Ex 2

01000011 01100101 01110100 00100000 01100101 01111000 01100101 01110010 01100011
 01101001 01100011 01100101 00100000 01100101 01110011 01110100 00100000 01110101
 01101110 00100000 01110000 01100101 01110101 00100000 01100110 01100001 01110011
 01110100 01101001 01100100 01101001 01100101 01110101 01111000 00101110

Ex 3

1. $496_{(16)}$
2. $3E_{(16)}$

3. $8D75_{(16)}$
4. $3C03_{(16)}$

Ex 4

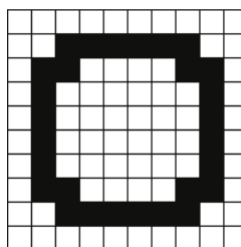
On commence par découper la suite de bits en octets : 01000011 00100111 01100101 01110011
 01110100 00100000 01100110 01100001 01100011 01101001 01101100 01100101.

Chaque octet représente un nombre entier : 67, 39, 101, 115, 116, 32, 102, 97, 99, 105, 108, 101. On cherche ensuite la table des codes ASCII en ligne de manière à traduire chacun de ces nombres en une lettre : « C'est facile ».

Ex 5

« 0tett1 »

Ex 7



Ex 8

a) 0 0 0 0 0 0
 0 1 1 1 1 0
 0 0 0 1 0 0
 0 0 0 1 0 0
 0 1 1 1 0 0
 0 0 0 0 0 0

b) 0 0 0 0 1 0
 0 0 0 0 1 1
 0 1 1 1 1 0
 0 1 0 1 0 0
 0 1 0 1 0 0

0 0 0 0 0 0

c) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0

0 0 1 1 0 1 1 1 0 1 1 0 0
 0 1 1 1 1 1 1 1 1 1 1 1 0
 0 1 0 1 1 1 1 1 1 1 0 1 0
 0 1 0 1 0 0 0 0 0 1 0 1 0
 0 0 0 0 1 1 0 1 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0

Ex 9