

ENCODAGE DE L'INFORMATION

CORRECTION

1. La serre bioclimatique

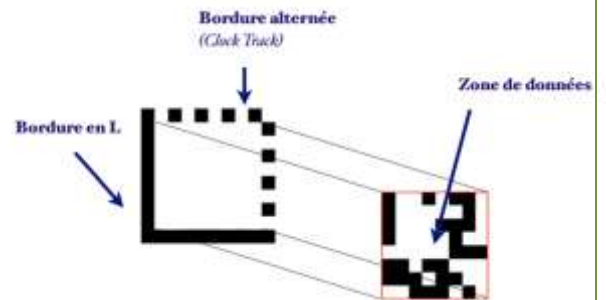
Dans son plan de développement, le maraîcher aimerait ouvrir ses serres au public afin de proposer la cueillette directement sur plan. Pour être toujours à la pointe de la technologie, le maraîcher a décidé de mettre des étiquettes devant chaque rangée de plantation complétée d'un code qui donnerait le nom de la plante. Ainsi chaque cueilleur pourrait flasher le code afin de retrouver le nom.



Après avoir éditer ces codes via une application en ligne, il a en sa possession tous les codes de toutes ses plantations. Tout en parcourant ses serres, il flashe les codes les uns après les autres et installe les étiquettes au bon endroit. N'ayant plus de batterie sur son smartphone pour les deux dernières étiquettes, il n'est pas en mesure de reconnaître quel code correspond à quelle plante. Peux-tu l'aider ?

La première étape consiste à mettre le code dans le bon sens en repérant la bordure en forme de L.

On peut ainsi déduire la zone de données (8*8).



Sur une matrice simple (8*8), le code est constitué de 8 caractères. Le découpage sur la grille de lecture d'une matrice simple (8*8) est donné ci-dessous.

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6

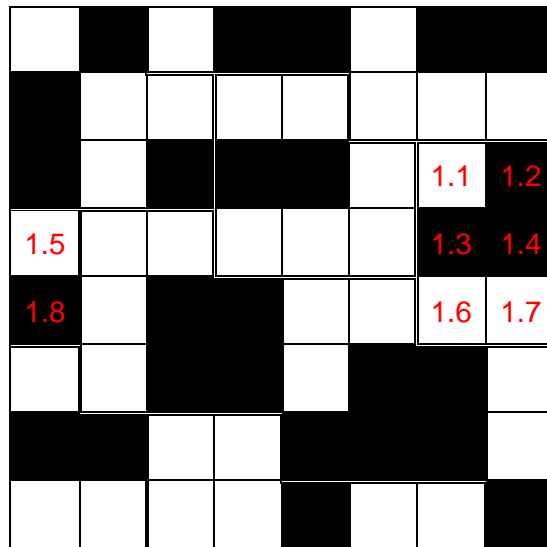
Le décodage du 1^{er} caractère du code s'effectue en analysant les cases 1 encadrées (1.1 ; 1.2, ...) contenant des 0 (case blanche) ou 1 (case noire). Et ainsi de suite pour les 6 autres caractères. ...



Voici la première étiquette. Le premier caractère du code (case 1) correspond à la première lettre du nom de la plante. Pouvez-vous décoder ce premier caractère afin d'aider le maraîcher ?

BETTERAVE ou POMME DE TERRE

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6



Q1- Donner le code en binaire (codework) **0 1 1 1 0 0 0 1**

Q2- Transformer ce codework en valeur décimale (codage sur 8 bits comme dans l'exemple ci-dessous.) et donner le code décimal Exemple :

0	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

0 1 1 1 0 0 0 1	$= 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 113$
	Pour les valeurs décimales comprises entre 0 et 127 , on enlève 1 soit 112 , puis on cherche le caractère correspondant dans la table ASCII: p
	Les valeurs décimales comprises entre 130 et 229, sont utilisées pour coder les 100 paires de 00 à 99.

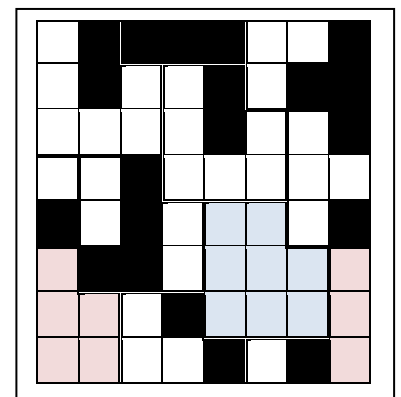
Q3- A l'aide du tableau ASCII donné ci-dessous, retrouver le caractère correspondant :

Alors betterave ou pomme de terre ?

Pomme de terre car « p »

Q4-Pour accueillir les clients, le maraîcher a construit le code présenté à droite qui donne le message suivant : BONJOUR!

Il lui manque le codage des deux derniers caractères (R !) qui se sont effacés, pouvez-vous lui donner le code binaire correspondant aux deux caractères R et ! puis compléter son code que vous reproduirez sur votre feuille.



Le code ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

D'après la table ASCII ci-contre la lettre « R » correspond au chiffre 82 auquel on rajoute 1 pour avoir une bonne transformation du code donc il faudra convertir 83 en binaire.

83/2=41 reste 1

41/2=20 reste 1

20/2=10 reste 0

10/2=5 reste 0

5/2=2 reste 1

2/2=1 reste 0

1/2=0 reste 1

Sens de lecture ↑

Le code est : **0 1 0 1 0 0 1 1**

Le zéro est à ajouter au début pour avoir un octet soit 8 bits, ce qui ne changera pas la valeur décimale

On fait la même chose pour le « ! » :

On convertit la valeur 34 car « ! » correspond à 33 dans la table ASCII.

34/2=17 reste 0

17/2=8 reste 1

8/2=4 reste 0

4/2=2 reste 0

2/2=1 reste 0

1/2=0 reste 1

Sens de lecture ↑

Donc le code est **0 0 1 0 0 0 1 0**

