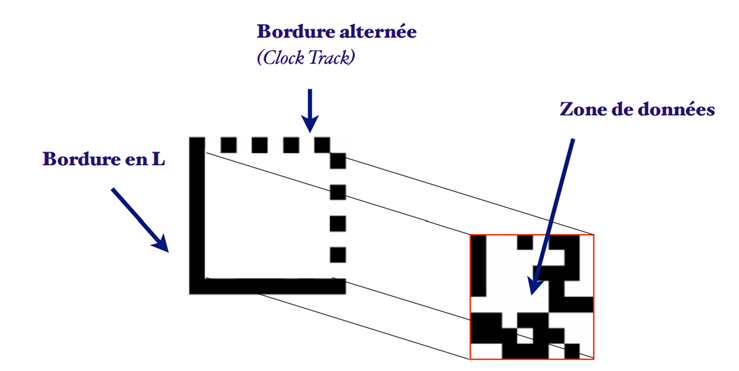
ENCODAGE DE L’INFORMATION

**CORRECTION**

# La serre bioclimatique

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

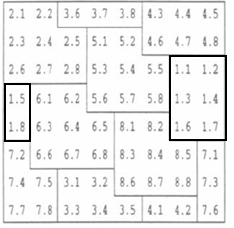
****

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 reconnaitre 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.

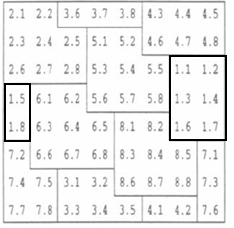


Le décodage du 1° caractère du code s’effectue en analysant les cases 1 encadrées (1.1 ; 1.2, …) contenant des 0 (case blanche) ou 1 (case noire). Et ainsi de suite pour les 6 autres caractères. …

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 maraicher ?

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 1.1 | 1.2 |
| 1.5 |  |  |  |  |  | 1.3 | 1.4 |
| 1.8 |  |  |  |  |  | 1.6 | 1.7 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

***BETTERAVE ou POMME DE TERRE***



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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 0 | 1 | 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 1 0 0 0 1** | =**0**\*27+**1**\*26+**1**\*25+**1**\*24+**0**\*23+**0**\*22+**0**\*21+**1**\*20= **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 :

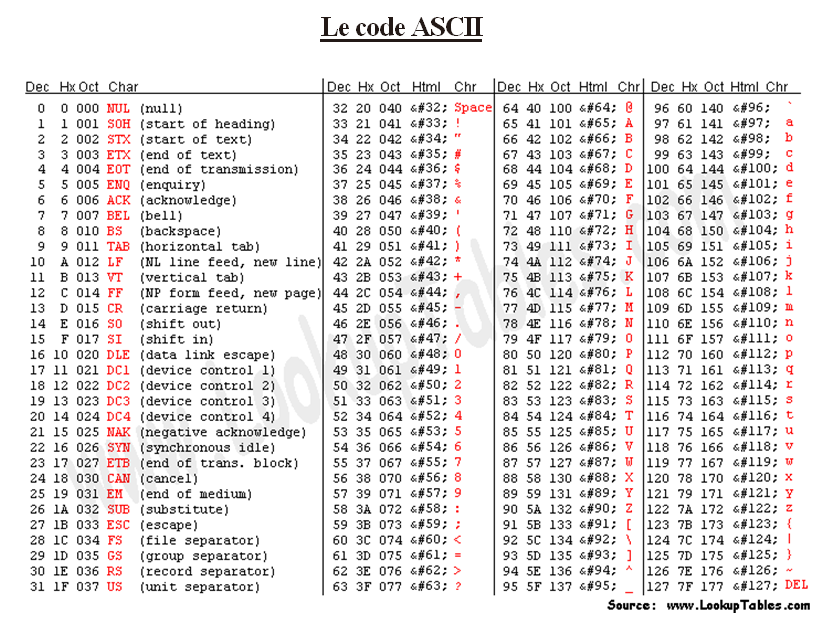
*Alors betterave ou pomme de terre ?*

**Pomme de terre car « p »**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Q4-Pour accueillir les clients, le maraicher 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.



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.

Sens de lecture

83/2=41 reste **1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  | 8.1 | 8.2 |  |  |
| 7.2 |  |  |  | 8.3 | 8.4 | 8.5 | 7.1 |
| 7.4 | 7.5 |  |  | 8.6 | 8.7 | 8.8 | 7.3 |
| 7.7 | 7.8 |  |  |  |  |  | 7.6 |

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**

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**

Sens de lecture

8/2=4 reste **0**

4/2=2 reste **0**

2/2=1 reste **0**

1/2=0 reste **1**

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