- TP1 : Conception d'un éditeur de codes à barres -

Le code à barres ou "code-barres" permet d'identifier automatiquement et très rapidement des produits par une simple lecture optique. Utilisée dans plus de 150 pays, la version de code à barres utilisée en France dans les supermarchés est actuellement diffusée par l'organisation internationale $GS1$ qui se charge notamment de l'attribution des codes. Né d'une idée de Joe Woodland en 1948 à Miami, le code à barres a été utilisé massivement à partir des années 1970 aux USA et au Canada. Actuellement, il y a 13 milliards de lectures de code à barres par jour. Ce TP propose de concevoir un éditeur de code à barres à partir du numéro EAN.

1 Principe du codage

Les codes à barres utilisés en France sont au format EAN (*European Article Numbering*). Il s'agit d'une succession de traits noirs et blancs d'épaisseur variable, qui permet de coder une série de chiffres selon une norme bien précise. La plupart des numéros EAN possèdent 13 chiffres (figure 1) mais certains seulement 8 (figure 1). Dans un souci de simplicité, nous nous éditerons des codes à barres à 8 chiffres d'une part et vérifierons la clé de contrôle des codes à 13 chiffres d'autre part.

Figure 1 – code à barres selon la norme EAN

Les traits noirs et blancs d'épaisseur variable du code à barres sont en fait constitués d'une suite de traits élémentaires noirs ou blancs ayant tous la même épaisseur. Par exemple, deux traits noirs consécutifs permettent d'obtenir un trait noir deux fois plus épais. Chaque chiffre du numéro EAN est codé par une succession de 7 traits élémentaires, les tables de conversion étant représentées sur la figure 2, où un trait blanc est représenté par le caractère "_" et un trait noir par "X". Suivant sa position dans le code, un chiffre est codé selon le tableau de conversion en élément de type A, B ou C.

Dans la programmation sous *Python*, on décide plutôt de coder un trait noir élémentaire par l'entier 1 et un trait blanc par l'entier 0. Par ailleurs, la succession des traits sera représentée par une liste. Par exemple, le chiffre 4 codé en élément de type A sera représenté par la liste $[0, 1, 0, 0, 0, 1, 1]$.

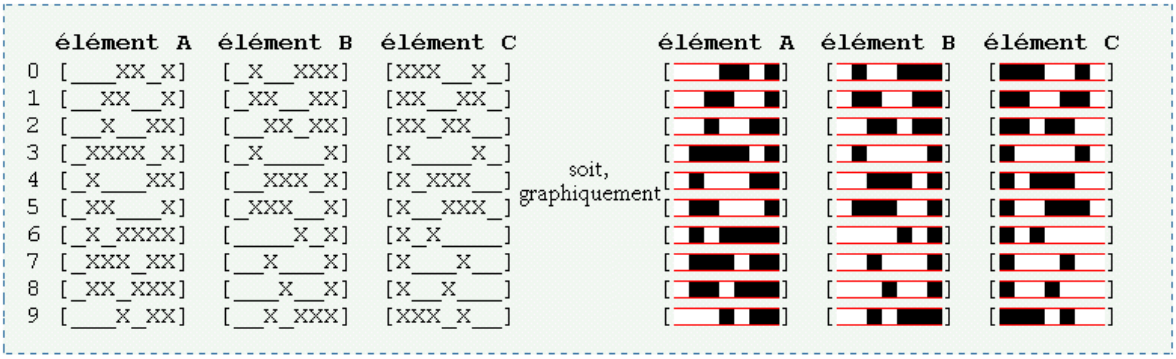

Figure 2 – table de conversion EAN

Dans le codage *EAN* 8, le code à barres est constitué successivement (cf figure 3) :
— d'une zone de garde initiale, représentée par [1,0,1],
— d'une zone codant la première moitié des chiffres sous forme d'éléments de type A,
— d'une zone de garde centrale, représentée par [0,1,0,1,0],
— d'une zone codant la deuxième moitié des chiffres sous forme d'éléments de type C.
— d'une zone de garde finale, représentée par [1,0,1].

La figure 3 suivante, représente un code à barres à 12 chiffres selon la norme *EAN.UCC*-12, identique à la norme *EAN* 8 avec 12 chiffres au lieu de 8 (cette norme est utilisée entre autres aux États-Unis, mais pas en France). On peut y distinguer les trois zones de garde surlignées en vert ainsi que le codage détaillé du chiffre 4 en élément de type C.

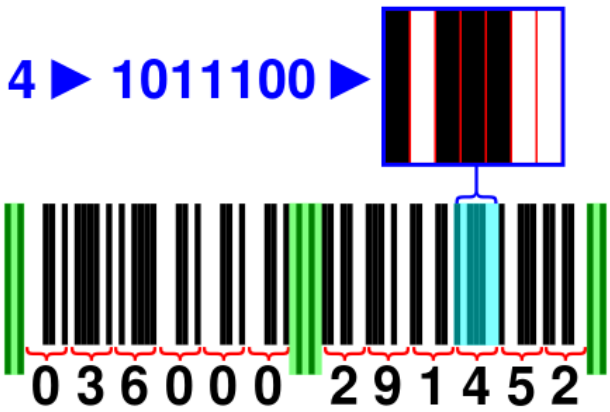

Figure 3 – principe du codage *EAN*

2 Questions préliminaires/Rappels

Les questions suivantes portent sur des outils servant à la création de l'éditeur de codes à barres.

1. Indiquer le lien simple entre la table de conversion en élément A et celle en élément C.
2. Écrire une formule unique d'affectation du type $a = ...$ remplaçant "si a vaut 0, alors a prend la nouvelle valeur 1, sinon, si a vaut 1, alors a prend la nouvelle valeur 0".
3. Rappeler la syntaxe *Python* pour créer une boucle pour laquelle la variable i est égale successivement aux entiers de 0 à 6 inclus.

4. Rappeler comment obtenir la longueur d'une chaîne de caractères ou d'une liste.
5. Rappeler la syntaxe pour récupérer le caractère de rang i d'une chaîne de caractères (par exemple, si num='85679', comment récupérer le 2^e chiffre de num en partant de la gauche ?)
6. Rappeler une syntaxe possible pour concaténer deux listes (par exemple, comment créer [4,5,6,7,8] à partir de [4,5,6] et [7,8]).
7. Rappeler la ligne de commande à écrire pour importer le module permettant le tracé de graphiques, ainsi que les instructions pour créer une figure en lui donnant un nom. Préciser également la commande pour y enlever les axes et enfin celle pour afficher la figure.

3 Table de conversion A

▼ Script 1

Créer une fonction `conversionA` ayant comme paramètre un caractère (type *string*) et qui :
- si le caractère correspond à un chiffre ('0','1',... ou '9'), renvoie la liste de 0 et de 1 selon la table de conversion en élément de type A,
- sinon, renvoie une liste vide (qui sera à interpréter par l'utilisateur comme le message d'erreur : "un des caractères du code n'est pas un chiffre").

A titre d'illustration, l'utilisation de cette fonction dans le prompt donnera :

```
>>> conversionA('4')
[0,1,0,0,0,1,1]
>>> conversionA(4)
[]
```

4 Table de conversion *EAN* 8

▼ Script 2

Créer une fonction nommée `conversionEAN8` ayant comme paramètre un caractère et le rang du chiffre à coder (par convention, le rang du premier chiffre est zéro) :
- si le caractère correspond au chiffre ('0','1',... ou '9'), elle renvoie la liste de 0 et de 1 selon la table de conversion :
  - en élément de type A si le rang du chiffre est inférieur ou égal à 3,
  - en élément de type C si le rang du chiffre est strictement supérieur à 3.
- sinon, elle renvoie la liste vide.

IMPORTANT : Dans la fonction *conversionEAN8*, il est demandé de faire appel **exclusivement** à la fonction *conversionA* pour générer la liste, quitte à la modifier si besoin suivant le rang du chiffre.

A titre d'illustration, l'utilisation de cette fonction dans le prompt donnera :

```
>>> conversionEAN8('4',2)
[0,1,0,0,0,1,1]
>>> conversionEAN8('4',5)
[1,0,1,1,1,0,0]
```

5 Création de la liste binaire

▼ Script 3

Créer une fonction nommée `creationliste` de paramètre une chaîne de caractères et retournant la liste binaire représentant l'ensemble des traits noirs et blancs élémentaires formant le code à barres selon la norme *EAN* 8. La chaîne de caractères mis en argument devant correspondre à un numéro *EAN* 8, un message d'erreur spécifique devra être affiché si cette chaîne ne possède pas 8 caractères.

A titre d'illustration, pour la chaîne '12345678', la fonction doit retourner :

[1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1]

6 Affichage du code à barres

▼ Script 4

Créer une fonction nommée `affichage` ayant comme paramètre une liste binaire (constituée de 0 et de 1) de taille quelconque et qui affiche dans une figure appelée 'code à barres' le code à barres correspondant à la liste (1 représentant un trait noir et 0 une absence de trait). Pour tracer un trait noir élémentaire, on pourra utiliser la fonction *axvspan* qui permet de tracer un rectangle vertical noir entre les abscisses x et $x+d$ selon la syntaxe : `axvspan(x,x+d,facecolor='0')`

7 Editeur de codes à barres *EAN* 8

▼ Script 5

Utiliser l'ensemble des fonctions précédentes pour créer une fonction nommée `editeurEAN8` sans paramètre d'entrée, mais qui récupère la chaîne de caractères saisie au clavier par l'utilisateur et affiche le code à barres correspondant au numéro saisi selon la norme *EAN* 8.

Pour l'étape de la saisie, on pourra utiliser la fonction *input* sous la forme :

```
numero=input('Entrer le code EAN à 8 chiffres : ')
```

8 Vérification de la clé de contrôle

Pour les codes *EAN* à 13 chiffres, le dernier chiffre est une clé de contrôle : calculé en fonction de tous les chiffres précédents, il permet de sécuriser le code en détectant une éventuelle erreur de lecture ou de saisie. Le calcul de la clé est le suivant : on somme les six chiffres de rang pair (rappel : le rang est compté de gauche à droite en partant de zéro). On ajoute à cette somme le triple de la somme des six chiffres de rang impair. On regarde le chiffre des unités du résultat obtenu. Si ce chiffre est zéro, la clé vaut 0. Sinon, la clé vaut la différence entre 10 et ce chiffre.

A titre d'exemple, pour le code '123456789123X' où X désigne la clé, la somme des six chiffres de rang pair vaut $1+3+5+7+9+2=27$. On ajoute $3\times(2+4+6+8+1+3)=72$ et l'on obtient 99. La clé vaut donc $X=10-9=1$.

▼ Script 6

Créer une fonction nommée `verifcle` ayant comme paramètre une chaîne de caractère et retournant le booléen indiquant si la chaîne de caractères en entrée correspond à un code *EAN* 13 ayant une clé correcte. On pourra tester la fonction avec le numéro du code de la figure 1 et sur ceux d'objets de la vie courante.

Pour l'implémentation de la fonction, on rappelle que pour convertir un caractère noté a en entier (transtypage), on peut écrire *int(a)*.