

## Séance 3: JEUX DE CARACTÈRES

L1 – Université Nice Sophia Antipolis

### Exercice 1 (Échauffement, ★)

Au top-level,

1. faites afficher le code ASCII des caractères 'A', 'D', '0' (zéro), '3' puis vérifiez sur la table ASCII page suivante; voyez-vous comment on peut déduire le code ASCII de toute lettre majuscule à partir de celui de 'A', et de tout chiffre à partir de celui de '0' ?
2. faites calculer à Python la position dans l'alphabet de la lettre M (réponse : 12 si on compte à partir de 0, sinon 13 pour les humains);
3. faites afficher à Python les caractères de code ASCII compris entre 32 et 44 sur une même ligne (réponse : `! " # $ % & ' ( ) * + , .`).
4. Définissez une fonction `est_chiffre(c)` qui renvoie True si le caractère `c` est un chiffre. Vous n'utiliserez pas la méthode `isdigit`, ce serait de la triche. Bon, ou alors uniquement une fois que vous avez vu comment faire sans.
5. Définissez une fonction `masque_numero(s)` qui renvoie la chaîne `s` dans laquelle les chiffres sont remplacés par des étoiles. Testez au toplevel.

```
>>> masque_numero('Bonjour je vends 1 chat. Appelez au 0678912345.')
'Bonjour je vends * chat. Appelez au *****.'
```

□

### Exercice 2 (Cryptographie antique, ★★)

Un système cryptographique ancien, souvent appelé code de César, consiste à choisir une clé entière `k` entre 1 et 25 pour fabriquer, à partir d'un message `msg`, un nouveau message codé avec la technique suivante. Chaque lettre majuscule de `msg` est décalée de `k` positions vers la droite (l'alphabet est circulaire : après 'Z' on revient sur 'A'). Les autres caractères du message sont laissés intacts !

1. Programmez la fonction `code_cesar1(c,k)` qui retourne le caractère correspondant au chiffrement de `c`.

```
>>> code_cesar1('A', 3)
'D'
>>> code_cesar1('Z', 3)
'C'
```

```
>>> code_cesar1(' ', 3)
' '
>>> code_cesar1('a', 3)
'a'
```

2. Programmez la fonction `code_cesar(msg,k)` qui retourne le message codé, en appliquant la transformation précédente à chaque caractère.

```
>>> code_cesar('ENVOYEZ 36 HOMMES !',3)
'HQYRBHC 36 KRPPHV !'
```

3. Sur le même modèle, programmez la fonction `decode_cesar(msg,k)` qui prend un message codé et retourne le message en clair.
4. Défi urgent : décidez le message 'JLGVl\_XRJFZC' dont César a perdu la clé !

□

**Exercice 3 (Table ASCII, \*\*)**

Écrivez un programme qui affiche la table ASCII ci-dessous en respectant la mise en page.

32	33 !	34 "	35 #	36 \$	37 %	38 &	39 ' ,
40 (	41 )	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [	92 \	93 ]	94 ^	95 _
96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124	125 }	126 ~	

□

**Exercice 4 (Print Proust, \*\*\*)**

Écrivez une fonction `justify_left(s,n)` qui prend une chaîne de caractères `s` contenant une phrase (sans aucun caractère de contrôle) et qui l'affiche à l'écran en mettant au plus `n` caractères par ligne, en justifiant à gauche.

```
>>> justify_left('Mais au lieu de la simplicité, c'est le faste que...', 12)
Mais au lieu
de la
simplicité
c'est le
faste que...
```

Testez votre programme en affichant cette phrase de Proust en entier sur 80 caractères par ligne (demandez la phrase à Google).

□

### Exercice 5 (Attaque sur un code de sécurité sociale, \*\*-\* \*\*)

On se propose d'étudier une méthode pour crypter et décrypter un message  $m$  à l'aide d'une clé  $k$ . Le message  $m$  et la clé  $k$  sont des chaînes de caractères qui ne comportent que les caractères '0' et '1'. L'opération à la base du procédé est le ou exclusif, noté  $\oplus$ , et défini comme suit : si  $c_1$  et  $c_2$  sont des caractères distincts,  $c_1 \oplus c_2$  est le caractère 1, sinon c'est le caractère 0.

1. Écrivez une fonction `chr_xor(c1,c2)` qui prend en arguments deux caractères  $c_1$  et  $c_2$  et qui renvoie le caractère  $c_1 \oplus c_2$ . Par exemple, `chr_xor('0','1')` renvoie '1'.
2. L'encryption  $e$  de  $m$  avec la clé  $k$  est la chaîne de caractères de la même longueur que  $m$  dont le  $i$ -ème caractère est le ou exclusif entre le  $i$ -ème caractère de  $m$  et le  $i$ -ème caractère de  $k$ ; si  $m$  est plus long que  $k$ , on répète la clé  $k$  à la suite d'elle-même pour obtenir une chaîne de caractères de la même longueur que  $m$ . Par exemple, si la clé est '01' et que  $m$  comporte 5 caractères, on rallonge  $k$  en '01010'.

Écrivez une fonction `encrypt(m,k)` qui renvoie l'encryption de  $m$  avec la clé  $k$ . Par exemple, on aura `encrypt('1110011','10')` renvoie '0100110'.

3. On veut utiliser ce schéma d'encryption pour crypter un numéro de sécurité sociale comportant 15 chiffres décimaux. Pour cela, il suffit de convertir le numéro de sécurité sociale en une chaîne de caractères 0 ou 1, puis d'appliquer la fonction `encrypt`. Écrivez une fonction `to_bin(ss_id)` qui fait cette première étape : l'argument `ss_id` est une chaîne de caractères contenant uniquement des chiffres de 0 à 9, et la fonction renvoie la suite des écritures de ces chiffres en base 2, chacun sur 4 bits. Par exemple, `to_bin('0123')` renvoie '0000000100100011' renvoie '0000'+'0001'+'0010'+'0011'.

4. Pour vérifier si un numéro de sécurité sociale est valide, on additionne le nombre  $n_1$  formé par les 13 premiers chiffres au nombre  $n_2$  formé par les 2 derniers chiffres, et on vérifie que c'est un multiple de 97. Par exemple, le numéro 2 55 08 14 168 025 38 est un numéro de sécurité sociale valide car  $2550814168025 + 38 = 26297053279 \times 97$ .

Écrivez une fonction `is_valid_ssid(s)` qui prend en argument une chaîne de caractères  $s$  et qui renvoie True si  $s$  est un numéro de sécurité sociale valide. Par exemple, `is_valid_ssid('255081416802538')` renvoie True.

5. Vous avez intercepté le numéro de sécurité sociale crypté suivant '1010111101011011011011111011101111110000010101010100111111010' et vous savez que la personne à qui il appartient est un homme né en janvier 98 dans les Alpes maritimes – autrement dit, le numéro de sécurité sociale commence par '1980106'. Enfin, vous savez que le message est crypté avec une clé  $k$  sur 32 bits. Saurez-vous retrouver le numéro de sécurité sociale ainsi que la clé  $k$  ?

□