Représentation d'un texte en machine

0 NUL	32 espace	64 @	96 `
1 SOH	33 !	65 A	97 a
2 STX	34 "	66 B	98 b
3 ETX	35 #	67 C	99 c
4 EOT	36 $	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 '	71 G	103 g
8 BS	40 (	72 H	104 h
9 HT	41)	73 I	105 i
10 LF	42 *	74 J	106 j
11 VT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 SLE	48 0	80 P	112 p
17 CS1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21 NAK	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EM	57 9	89 Y	121 y
26 SIB	58 :	90 Z	122 z
27 ESC	59 ;	91 [	123 {
28 FS	60 <	92 \	124 |
29 GS	61 =	93]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 ■

Une table de conversion associe à chaque caractère ou opérateur, un nombre décimal, lui-même converti en binaire. Par exemple :

A la lettre B est associé le nombre 66 …

A la lettre a est associé le nombre 97, à b est associé le nombre 98 …

1. Créer un algorithme en Python qui permet d'afficher la table ASCII de la manière suivante, c'est-à-dire sur chaque ligne, le code ASCII suivi du caractère correspondant.

…..

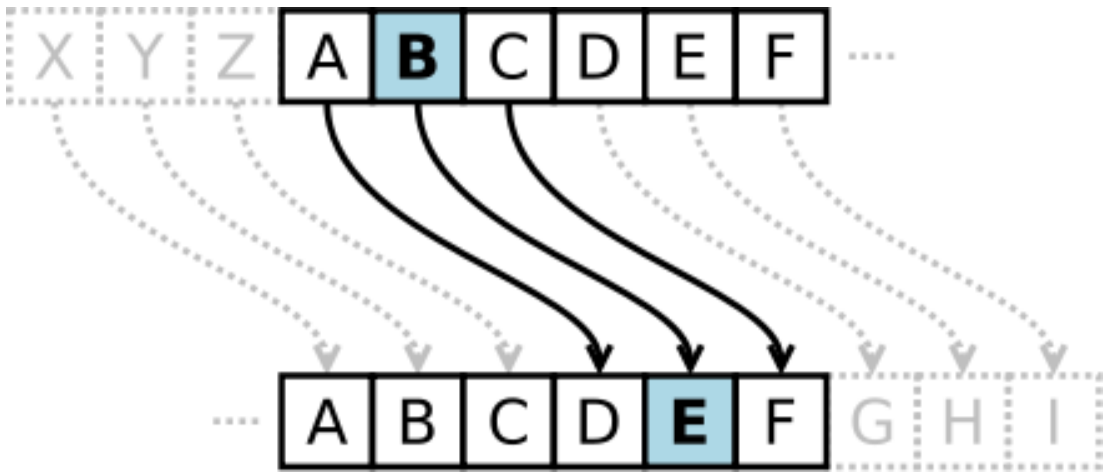
33 !
34 "
35 #
36 $
37 %

….

Des indices : Passage de caractère au code ASCII et inversement print(ord("b")) print(chr(97))

2. **Application de l'utilisation de la table ASCII à la cryptographie.**

Le cryptage César :



Objectif du jeu: Ouvrir le coffre !

Sur picassciences, vous trouverez le lien vers un coffre-fort virtuel. Entrez votre prénom et cliquez sur **Valider**.

Testez vos résultats pour savoir si vous êtes sur la bonne voie.

Dans ce coffre, vous pouvez entrer vos résultats, l'application pourra vous dire s'ils sont bons, ou s'il y a des erreurs. La réponse finale débloquera le coffre !

Débloquer des indices sur la démarche à suivre

S'il vous manque un élément sur la démarche à suivre, vous trouverez des indices pour vous guider dans l'activité. L'objectif est d'arriver à ouvrir le coffre-fort en utilisant le moins d'indices possibles. Mais si vous êtes bloqué(e), il faudra prendre un indice.

Tapez dans le champ de texte (sans les guillemets et en minuscule)

« indice 1 » pour vous aider à bien commencer le problème sur la cryptographie César

« indice 2 » pour vous aider à saisir les problématiques intermédiaires sur le problème sur la cryptographie César

« indice 3 » pour vous aider à saisir à conclure sur le problème sur la cryptographie César

« indice 4 » pour vous aider à faire le décryptage César

« indice 5 » pour vous aider à bien commencer le problème sur la cryptographie Enigma

« indice 6 » pour mettre en avant un détail qui peut vous bloquer sur la cryptographie Enigma

1. Compléter le script python suivant pour crypter une chaine de caractère selon la méthode de César :

```
Mot = "Picassciences"
Cle = 2 # on utilise un entier ici
Code = "" # la variable du mot crypté

for i in range(0,len(Mot)):

     Code = Code + …… (partie en pointillés à mettre dans le coffre sans espace)

print(Code)
```

2. Créer un algorithme pour décoder un message selon cette méthode (peu de modifications sont nécessaires)

3. Le cryptage Enigma (simplifié) :

Le cryptage Enigma reprend le principe de César, sauf que la clé numérique est aussi longue que le mot à coder. Il était utilisé pendant la seconde guerre mondiale par les allemands pour communiquer. Chaque jour, une nouvelle clé était utilisée, ce qui rendait le décryptage très difficile.

1. Adaptez les algorithmes précédents pour créer un algorithme de cryptage Enigma. On se limitera à des décalages allant de 0 à 9 pour chaque lettre.

Ici on décale b *de 1 rang,* o *de 2 rangs,* n *de 3 rangs,* j *de 4 rangs, etc…*

```
Mot = "bonjour"
Cle = "1234567" #on utilise une chaine de caractère
Code = "" #la variable du mot crypté

for i in range(0,len(Mot)):

      Code = Code + …… (partie en pointillés à mettre dans le coffre sans espace)
```

Des indices : Pour passer d'un type de variable à l'autre

Transformer une chaine de caractère en entier : int("2")

Transformer un nombre entier en chaine de caractère : str(42)

Transformer une chaine de caractère en flottant : float("42")

Créer un algorithme pour décoder un message selon cette méthode (peu de modifications sont nécessaires)

```
Mot = "Cqqnt{y"
Cle = "1234567" #on utilise une chaine de caractère
Code = "" #la variable du mot crypté

for i in range(0,len(Mot)):

      Code = Code + …… (partie en pointillés à mettre dans le coffre sans espace)
```

4. Chiffrement Vigenère :

Le **chiffre de Vigenère** est un système de chiffrement polyalphabétique, c'est un chiffrement par substitution, mais une même lettre du message clair peut, suivant sa position dans celui-ci, être remplacée par des lettres différentes, contrairement à un système de chiffrement mono alphabétique comme le chiffre de César (qu'il utilise cependant comme composant). Cette méthode résiste ainsi à l'analyse de fréquences, ce qui est un avantage décisif sur les chiffrements mono alphabétiques. Cependant le chiffre de Vigenère a été percé par le major prussien Friedrich Kasiski qui a publié sa méthode en 1863. Depuis cette époque, il n'offre plus aucune sécurité.

Consulter la vidéo sur picassciences pour comprendre le fonctionnement.

On part de l'algorithme Enigma, sur lequel on procède à des modifications. On utilise une clé de 4 chiffres comme suit :

```
Mot = "voilaunephrasemultipledequatrecaractères"

Cle = "1234"

Code = ""

for i in range(0,len(Mot)):

    Code = Code + chr(ord(Mot[i])+ int(Cle[i]) )

print(Code)
```

Apporter des modifications pour faire fonctionner le chiffrement Vigenère