BREVET DE TECHNICIEN SUPÉRIEUR

SERVICES INFORMATIQUES AUX ORGANISATIONS

SESSION 2012

Sous-épreuve E 22 Algorithmique appliquée

- Il est formellement interdit d'utiliser toute connexion à un réseau interne ou externe au centre d'examen, quel qu'en soit le procédé.
- L'usage d'une calculatrice est autorisée.
- Cette épreuve comporte 2 parties :
  - une première partie qui dure 30 minutes à l'issue de laquelle vous devez fournir une production écrite répondant au sujet.
  - une deuxième partie qui dure également 30 minutes. Vous travaillerez sur un des ordinateurs d'examen pour coder les algorithmes papiers sous Python.
    Vous enregistrerez votre travail sur une clé USB fournie par votre professeur.
    Vous imprimerez votre travail et le signerez.
- Vous n'oublierez pas de rendre :
  - Le sujet.
  - L'algorithme papier où figureront votre nom et prénom.
  - L'impression de votre programme signée

Nom : **Prénom :**

Problème :

Pour coder un message, on procède de la manière suivante : à chacune des 26 lettres de l'alphabet, on commence par lui associer un entier n de l'ensemble
$\Omega = \{0\ ;\ 1\ ;\ 2\ ;\ \ldots\ ;\ 24\ ;\ 25\}$ selon le tableau ci-dessous :

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Puis on associe à tout entier n de Ω le reste de la division euclidienne de $(2n + 3)$ par 26 ; ce reste est alors associé à la lettre correspondante.

Exemple : pour coder la lettre P on procède de la manière suivante :

Étape 1 : on lui associe l'entier $n = 15$.

Étape 2 : on calcule de $2 \times 15 + 3 = 33$.

Étape 3 : le reste de la division de 33 par 26 est 7.

Étape 4 : on associe 7 à H. Donc P est codé par la lettre H.

Les étapes 2 et 3 sont appelées codage affine.

L'algorithme et la traduction Python du programme principal sont :

Variables :

- alphabet,code_chiffre1, code_chiffre2 : liste d'entiers
- alphabet : liste de carctères
- mot, mot_code : chaîne de caractères

Début

```
alphabet=["A","B","C","D","E","F","G","H","I","J",
          "K","L","M","N","O","P","Q","R","S","T","U",
          "V","X","X","Y","Z"]

# Lecture du mot à coder

Lire le mot à coder, mot

# Codage du mot en chiffres

code_chiffre1 ⟵ code_mot_chiffre(mot)
Écrire code_chiffre1

# Codage affine

code_chiffre2 ⟵ code_chiffre_chiffre(code_chiffre1)
Écrire code_chiffre2

# Codage des chiffres au mot

mot_code ⟵ code_chiffre_mot(code_chiffre2)
Écrire mot_code
```

Fin

```
# Fonctions


# Programme principal

alphabet=["A","B","C","D","E","F","G","H","I","J",
          "K","L","M","N","O","P","Q","R","S","T","U",
          "V","X","X","Y","Z"]

mot=input("Entrer le mot à coder ")

# Codage du mot en chiffres

code_chiffre1=code_mot_chiffre(mot)
print(code_chiffre1)

# Codage affine

code_chiffre2=code_chiffre_chiffre(code_chiffre1)
print(code_chiffre2)

# Codage des chiffres au mot
mot_code=code_chiffre_mot(code_chiffre2)

print(mot_code)
```

Partie A : **8 points**

A traiter sur feuille et à rendre avant d'accéder aux machines.
Durée maximum : 30 minutes

Exercice 1

Écrire une fonction `code_mot_chiffre(mot)` dont le paramètre est la chaîne de caractères mot. Cette fonction renvoie le code chiffre.

Exemple : mot = "AMI" $\Longrightarrow$ `code_mot_chiffre(mot)`= [0, 12, 8]

Exercice 2

Écrire une fonction `code_chiffre_mot(code_chiffre)` dont le paramètre est la liste de chiffres `code_chiffre`. ette fonction renvoie la chaîne de caractères correspondante.

Exemple : `code_chiffre =[0, 12, 8]` $\Longrightarrow$ `code_chiffre_mot(code_chiffre)`="AMI"

Exercice 3

Écrire une fonction `code_chiffre_chiffre(code_chiffre)` dont le paramètre est la liste de chiffres `code_chiffre` qui renvoie le code chiffre en utilisant la formule : Reste de la division par 26 de $2 \times n + 3$, où n est le numéro de la lettre à coder.

Exemple : `code_chiffre= [0, 12, 8]` $\Longrightarrow$ `code_chiffre_chiffre(code_chiffre)`= [3, 1, 19]

Partie B : **8 points**

A traiter sur un ordinateur en utilisant le langage Python.
Vous enregistrerez votre travail sous votre nom.
Durée maximum : 30 minutes

Exercice 4

Coder sous Python la fonction `code_mot_chiffre(mot)` de l'exercice 1 puis la tester.

Exercice 5

Coder sous Python la fonction `code_chiffre_mot(code_chiffre)` de l'exercice 2 puis la tester.

Exercice 6

Coder sous Python la fonction `code_chiffre_chiffre(code_chiffre)` de l'exercice 3 puis la tester.

Exercice 7

Télécharger le programme principal. Faîtes un copier coller sur votre page contenant les fonctions puis le tester.

BREVET DE TECHNICIEN SUPÉRIEUR

SERVICES INFORMATIQUES AUX ORGANISATIONS

Aide mémoire

- a // b calcule le quotient entier de la division de a par b.
  Exemple : 13 // 5 donne 2
- a % b calcule le reste de la division de a par b.
  Exemple : 13 % 5 donne 3
- a / b calcule le quotient de a par b.
  Exemple : 13 // 5 donne 2.6
- a * b calcule le produit de a par b.
  Exemple : 3 * 5 donne 15
- a ** b calcule a à la puissance b.
  Exemple : 2 ** 3 donne 8
- Vrai et Faux sont les booléens `True` et `False`.
- `liste[i]` restitue l'élément de la liste de rang i.
  Exemple : Liste=[10,25,33,4,8] `Liste[1]` renvoie 25.
- Une matrice est une liste de listes.
  Exemple :
  - M=$[[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15]] \Longrightarrow \begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{matrix}$
  - M[0] est la première ligne de la matrice : [1,2,3,4,5]
  - M[1][2] est la valeur de la deuxième ligne, troisième colonne soit 8.
- `len(liste)` restitue la longueur de la liste.
  Exemple : Liste=[10,25,33,4,8], `Len(Liste)` renvoie 5.
- `Liste.append(a)` ajoute l'élément `a` à la liste `Liste`
  Exemple : Liste=[10,25,33,4,8], `Liste.append(100)` renvoie [10,25,33,4,8,100]
- `Liste.insert(i,a)` ajoute l'élément `a` au rang `i` de la liste `Liste`
  Exemple Liste=[10,25,33,4,8], `Liste.insert(0,100)` renvoie [100,10,25,33,4,8]
- `range(a,b)` est la liste des entiers compris entre a et b, b non compris.
  Exemple :
  - `range(0,5)` renvoie la liste [0,1,2,3,4]
  - `for i in range(0,5)` :
    .....
    correspond à : `Pour i allant de 0 à 4`
- Chaîne de caractères. Comme pour les listes :
  - `len(ch)` donne la longueur de la chaîne de caractères `ch`.
  - ch[0] est le premier caractère de la chaîne `ch`

  Exemple : ch="AMI", `len(ch)` renvoie 3. `ch[0]` renvoie A. ch +"S" renvoie "AMIS"
- Syntaxe pour l'écriture d'une fonction :
  Exemple :

```
>>>   def somme_produit(a,b):
          somme=a+b
          produit=a*b
          return somme,produit
```

donne

```
>>>   somme_produit(3,4)
(7, 12)
>>> a,b=somme_produit(3,4)
>>> a
7
>>> b
12
```