

NUMERIQUE et SCIENCES INFORMATIQUES

Épreuve de l'enseignement de spécialité

Sujet d'entraînement

Partie Pratique

Classe Terminale de la voie générale

Le candidat doit traiter les 2 exercices

Le sujet comporte 5 pages



Exercice 1 (4 points)

Dans une entreprise, vous êtes en charge de la réalisation d'un code barre formé de traits verticaux noirs espacés de blancs.

En codage noir et blanc, le noir est codé par le nombre 1 et le blanc par le nombre 0.

L'entreprise vous donne une succession de 0 et de 1 sous forme d'une liste et vous devez donner une image du code barre.

Par chance, une de vos collègues a déjà réalisé une partie du travail : elle a créé une fonction qui permet de tracer le code-barre en noir et blanc à partir d'une liste de listes de 0 et de 1.

Il ne vous reste plus qu'à créer une fonction qui crée la liste de listes voulus à partir de la liste de 0 et de 1 initialement donné par l'entreprise !

Ainsi, vous devez créer une fonction `barrer` qui prend en paramètre une liste `sequence` non vide formée de 0 et ou 1 et qui renvoie une liste de 12 listes de même taille telles que la première et la dernière soient entièrement remplies de 0, tandis que les autres soient identiques à la liste `sequence` saisie comme argument.

Exemples :

```
>>> barrer([1,0,0,1,0,1])
[[0, 0, 0, 0, 0, 0],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [1, 0, 0, 1, 0, 1],
 [0, 0, 0, 0, 0, 0]]
```

Exercice 2 (4 points)

Le code Morse est un code développé en 1832 pour transmettre un texte formé de lettres majuscules et de chiffres latins à l'aide d'impulsions courtes et longues.

Il est encore utilisé de nos jours dans l'armée.

Code morse international

1. Un tiret est égal à trois points.
2. L'espacement entre deux éléments d'une même lettre est égal à un point.
3. L'espacement entre deux lettres est égal à trois points.
4. L'espacement entre deux mots est égal à sept points.

A	• —	U	• • —
B	— • • •	V	• • — —
C	— — • • •	W	• — — —
D	— • • •	X	— • • — —
E	•	Y	— • — — —
F	• • — •	Z	— — — • •
G	— — — •		
H	• • • •		
I	• •		
J	• — — — —		
K	— • — — —	1	• — — — —
L	• — — • •	2	• • — — — —
M	— — —	3	• • • — — —
N	— — •	4	• • • • — —
O	— — — —	5	• • • • •
P	• — — — •	6	• • • • • •
Q	— — — • —	7	— — — • • •
R	• — — • •	8	— — — — • •
S	• • • •	9	— — — — — •
T	—	0	— — — — —

Dans cet exercice, nous nous limitons aux lettres majuscules où une impulsion courte sera notée avec le caractère tiret '-' tandis qu'une impulsion longue sera notée avec un chaîne de caractères formée de trois tirets '---' ; le caractère espace ' ' permet de séparer deux impulsions du codage d'une même lettre.

Par exemple, la lettre A codée par une impulsion courte suivie d'une impulsion longue sera codé dans cet exercice par la chaîne de caractère : '- ---'.

De plus, deux lettres différentes seront séparées d'une chaîne de caractères formée de **trois espaces** : ' ' ' '.

Ainsi, le mot AI sera codé par la chaîne de caractères : '- --- - -'.

Enfin, deux mots seront séparés par le caractère esperluète &.

Ainsi, la phrase J AI sera codée par la chaîne de caractères : '- --- --- ---&- --- - -'.

Voici un dictionnaire qui associe à chaque lettre majuscule son codage Morse :

```
dico_morse={
  'A': '- ---',
  'B': '--- - - -',
  'C': '--- - --- -',
  'D': '--- - -',
  'E': '- -',
  'F': '- - --- -',
  'G': '--- --- -',
  'H': '- - - -',
  'I': '- -',
  'J': '- --- --- ---',
  'K': '--- - ---',
  'L': '- --- - -',
  'M': '--- ---',
  'N': '--- -',
  'O': '--- --- ---',
```

```

'P': '- _ _ _ _ -',
'Q': '- _ _ _ _ - -',
'R': '- _ _ _ -',
'S': '- _ _ -',
'T': '- _ _ _',
'U': '- _ _ _ _',
'V': '- _ _ _ _ -',
'W': '- _ _ _ _ -',
'X': '- _ _ _ _ - -',
'Y': '- _ _ _ _ - -',
'Z': '- _ _ _ _ - -',
}

```

Compléter ci-dessous les fonctions `encoder`, `dico_inverse` et `decoder` :

- La fonction `encoder` prend en paramètre une chaîne de caractères `phrase` représentant une phrase formée seulement de lettres majuscules, d'espaces uniques et d'esperluètes & pour séparer chaque mot de la phrase et cette fonction renvoie la chaîne de caractères correspondant au codage en Morse de la phrase saisie.
- La fonction `dico_inverse` prend comme argument le dictionnaire `dico_morse` et elle renvoie le dictionnaire où les clés sont les chaînes de caractères correspondant au codage en Morse de chaque lettre tandis que les valeurs sont les lettres majuscules associées.
- la fonction `decoder` prend en paramètre une chaîne de caractères supposées correspondre à un texte codé en Morse et elle renvoie ce texte en clair, avec éventuellement un espace supplémentaire à la fin.

```

def encoder(phrase: str)-> str:
    """
    traduit un texte en majuscules en Morse
    avec un triple espace entre chaque lettre
    et un esperluète entre chaque mot.
    """
    phrase_codee = ""
    debut_mot = True # booléen qui sert à gérer l'ajout d'un triple espace entre deux caractères
    for lettre in phrase:
        if debut_mot: # cas d'une première lettre d'un mot
            phrase_codee = phrase_codee + ...
            debut_mot = False
        else:
            if lettre == " ": # cas d'un espace séparant deux mots
                phrase_codee = ...
                debut_mot = True
            else: # cas d'une lettre majuscule
                phrase_codee = phrase_codee + ...
    return phrase_codee

def dico_inverse(dico_morse)-> dict:
    """
    retourne un dictionnaire
    dont les clés sont des caractères morses
    et les valeurs sont les lettres associées
    """
    dico_inv = {}
    for ...:
        ...

```

```
return dico_inv
```

(suite du code page suivante)

```

def decoder(code:str)->str:
    """
    code est une chaîne de caractères formés d'éléments du code Morse.
    La fonction renvoie sous forme de chaîne de caractères le texte décodé correspondant à l'argument s
    """

    morse_dico = dico_inverse(dico_morse)
    decode = ""
    liste_mots = code.split("&") # permet d'obtenir dans liste_mots la liste des mots de la phrase cod
    for elt in liste_mots:
        liste_lettres = elt.split(" ") # permet d'obtenir dans liste_lettres la liste des lettres de
        for lettre in liste_lettres:
            if lettre != ' ':
                decode = decode + morse_dico[lettre]
        decode = decode + ...
    return decode

```

Vous pouvez tester la fonction `encoder` avec les phrases suivantes :

```

>>> encoder('SOS')
' _ _ _   _ _ _ _ _   _ _ _ _ '
>>>encoder('JE ME NOIE')
'_ - _ _ _ _ _   -&- - _ _ _   -&- - _   _ _ _ _ _   _ _ _ _ -'

```

Vous pouvez tester la fonction `encoder` avec les instructions suivantes :

```

>>> morse_dico=dico_inverse(dico_morse)
>>> morse_dico['--- --- -']
'G'

```

Vous pouvez tester la fonction `encoder` avec l'instruction suivante :

```

>>> decoder('- - - - &-   _ _ _   - - - - &- - _ _ _   - - - - - - - - - - - - - -')
' C EST BON '

```