

IngeSUP - TD 11 - Les structures de données 2

"Le travail éloigne de nous trois grands maux : l'ennui, le vice et le besoin."

Voltaire

Exercice 10.1

Considérons le dictionnaire suivant:

Entrée [30]:

```
mydict = {"device": "laptop", "constructeur": "acer", "ram": "8G", "processeur": "Intel core i5", "stockage": "500 G"}
```

1. Corrigez l'erreur "stockage": "750 G"

Entrée [31]:

```
mydict["stockage"]="750 G"
print(mydict)
```

```
{'device': 'laptop', 'constructeur': 'acer', 'ram': '8G', 'processeur': 'Intel core i5', 'stockage': '750 G'}
```

2. Créez un programme qui affiche la liste des clés, la liste des valeurs

Entrée [16]:

```
print("liste des clés :")
for i in mydict.keys():
    print(i)
print("\nliste des valeurs :")
for i in mydict.values():
    print(i)
```

liste des clés :
device
constructeur
ram
processeur
stockage

liste des valeurs :
laptop
acer
8G
Intel core i5
750 G

3. Inversez les paires "processeur": "Intel core i5" et "stockage": "500 G"

Entrée []:

```
# ? consignes pas claire sur ce qu'il faut faire comme "inversion"
```

4. Ajoutez la pair clé-valeur : "Système d'exploitation" : "Windows 10"

Entrée [17]:

```
mydict["Système d'exploitation"]="Windows 10"

print(mydict)
```

```
{'device': 'laptop', 'constructeur': 'acer', 'ram': '8G', 'processeur': 'Intel core i5', 'stockage': '750 G', 'Système d'exploitation': 'Windows 10'}
```

Exercice 10.2

On considère les trois dictionnaires Pythons qui regroupe la totalité du matériels informatiques:

Entrée [3]:

```
dicPC={"HP": 11 , "Acer": 7 , "Lenovo": 17 , "Del": 23}
dicPhone={"Sumsung": 22 , "Iphone": 9 , "Other": 13 }
dicTablette = {"Sumsung": 15 , "Other": 13}
```

1. Ecrire un programme qui regroupe la totalité du matériel informatique dans un seul dictionnaire

Entrée [20]:

```
dicPC={"HP": 11 , "Acer": 7 , "Lenovo": 17 , "Del": 23}
dicPhone={"Sumsung": 22 , "Iphone": 9 , "Other": 13 }
dicTablette = {"Sumsung": 15 , "Other": 13}
dicTOTAL={}
dicTOTAL.update(dicPC)
dicTOTAL.update(dicPhone)
dicTOTAL.update(dicTablette)

print(dicTOTAL)
#Le problème de cette méthode c'est que Le Samsung est écrasé
#on voudrait plutôt garder les 2 sous les noms SamsungPhone et SamsungTablette
dicPC={"HP": 11 , "Acer": 7 , "Lenovo": 17 , "Del": 23}
dicPhone={"Sumsung": 22 , "Iphone": 9 , "Other": 13 }
dicTablette = {"Sumsung": 15 , "Other": 13}

dicTOTAL={}
for i,j in dicPC.items():
    dicTOTAL[i+"PC"]=j
for i,j in dicPhone.items():
    dicTOTAL[i+"Phone"]=j
for i,j in dicTablette.items():
    dicTOTAL[i+"Tab"]=j

print(dicTOTAL)
#on peut aussi interpréter la question comme faire un dico de dico
dicTOTAL={"PC":dicPC, "Phone":dicPhone, "Tablette":dicTablette}
print(dicTOTAL)
#ce qui permet d'accéder aux éléments comme ça :
print(dicTOTAL["Phone"]["Sumsung"])
```

```
{'HP': 11, 'Acer': 7, 'Lenovo': 17, 'Del': 23, 'Sumsung': 15, 'Iphone': 9, 'Other': 13}
{'HPPC': 11, 'AcerPC': 7, 'LenovoPC': 17, 'DelPC': 23, 'SumsungPhone': 22, 'IphonePhone': 9, 'OtherPhone': 13, 'SumsungTab': 15, 'OtherTab': 13}
{'PC': {'HP': 11, 'Acer': 7, 'Lenovo': 17, 'Del': 23}, 'Phone': {'Sumsung': 22, 'Iphone': 9, 'Other': 13}, 'Tablette': {'Sumsung': 15, 'Other': 13}}
22
```

Exercice 10.3

Le braille est un système d'écriture tactile à points saillants, à l'usage des personnes aveugles ou fortement malvoyantes. Le système porte le nom de son inventeur Français : Louis Braille.

En braille standard :

- un caractère est représenté par six points numérotés de 1 à 6 et disposés comme le montre la figure 1 ;
- un point peut être saillant (en relief) ou non, comme le montre la figure 2 ;
- un caractère est défini par le nombre et la disposition des points en relief.

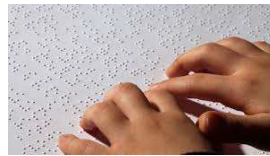


Figure (1)

Figure (2)

Le tableau suivant donne la représentation des 39 caractères de l'alphabet français. Dans cette représentation, chaque point rouge représente un point saillant.

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
•	•	•	•	•	•	•	•	•	•	•	•	•
à	â	ç	é	è	ê	ë	î	ô	œ	ù	û	ü
•	•	•	•	•	•	•	•	•	•	•	•	•

Pour simplifier le modèle, on représentera une lettre majuscule par le code suivant :

```

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 *_*_*

```

- La représentation d'un mot est une concaténation de blocs de six caractères.
- Chaque bloc de six caractères représente une lettre du mot.
- Un point saillant est représenté par un "*", et un point non saillant est représenté par un trait d'union "-".
- Les caractères "*" et "-" sont disposés selon l'ordre des numéros des points qu'ils représentent (c.f Figure 1).

1. À partir de la liste ci-dessous, écrivez le dictionnaire `latin2braille` ayant pour clé les 26 lettres majuscules de l'alphabet latin et pour valeur sa représentation braille.

►

* _ _ _ _ _


```
{'-----': ' ', '*-----': 'A', '-----': 'B', '*--*--': 'C', '-----': 'D', '*-----': 'E', '-----': 'H', '-----': 'G', '-
*--*--': 'I', '-*--*--': 'J', '-----': 'K', '-----': 'L', '*-----': 'M', '*-----': 'N', '*--*--': 'O', '-----': 'P', '-----
*--': 'Q', '-----': 'R', '-*--*--': 'S', '-*--*--': 'T', '*--*--': 'U', '-----': 'V', '-*--*--': 'W', '*--*--': 'X', '*--*--
*--': 'Y', '*--*--': 'Z', '-*--*--': '!'}
```

* _ _ * _ *** _ * _ ** _ * _ _ * _ _ _ _ _ *** _ * _ * _ * _ _ _ _ _ _ _ _ _ _ * _ _ _ * _ _ *** _ _ _ ** _ *

Entrée [37]:

```
texte="*--*--***-*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--"
```

```
def decode_braille(message):
    texteDécodé=""
    lettreBraille=""
    i=0
    while i<len(message):
        symbole=texte[i]
        lettreBraille += symbole
        if len(lettreBraille)==6:
            texteDécodé += braille2latin[lettreBraille]
            lettreBraille=""
        i+=1
    return texteDécodé

print(decode_braille (texte))

#alternative ou je découpe d'abord la liste en groupe de 6, dans une liste
#puis je traduit Lettre par Lettre

def decode_braille(message):
    L=[]
    lettre=""
    for i in message:
        lettre+=i
        if len(lettre)==6:
            L.append(lettre)
            lettre=''
    resultat=""
    print(L)
    for lettre in L:
        resultat+=braille2latin[lettre]

    return resultat

print(decode_braille (texte))
```

```
ESME RULES!
['*--*--', '*--*--', '*--*--', '*--*--', '*--*--', '*--*--', '*--*--', '*--*--', '*--*--', '*--*--']
ESME RULES!
```

Exercice 10.4

On considère le dictionnaire suivant dont les clés sont les noms des élèves et les valeurs des clés sont les notes obtenues en passant l'examen final:

Entrée [38]:

```
etudiants = {"etudiant_1" : 13 , "etudiant_2" : 17 , "etudiant_3" : 9 , "etudiant_4" : 15 ,
             "etudiant_5" : 8 , "etudiant_6" : 14 , "etudiant_7" : 16 , "etudiant_8" : 12 ,
             "etudiant_9" : 13 , "etudiant_10" : 15 , "etudiant_11" : 14 , "etudiant_112" : 9 ,
             "etudiant_13" : 10 , "etudiant_14" : 12 , "etudiant_15" : 13 , "etudiant_16" : 7 ,
             "etudiant_17" : 12 , "etudiant_18" : 15 , "etudiant_19" : 9 , "etudiant_20" : 17 }
```

1. Ecrire un programme Python qui partitionne ce dictionnaire en deux sous dictionnaires:

- **etudiantAdmis** dont les clés sont les étudiants admis et les valeurs des clés sont les moyennes obtenues (moyenne supérieurs ou égales à 10).
- **etudiantNonAdmis** dont les clés sont les étudiants non admis et les valeurs des clés sont les moyennes obtenues (moyenne inférieur ou égale à 10).

Entrée [27]:



```
etudiantAdmis={}
etudiantNonAdmis={}

for i,j in etudiants.items():
    if j>=10:
        etudiantAdmis[i]=j
    else:
        etudiantNonAdmis[i]=j

print("admis :", etudiantAdmis)
print("non admis :",etudiantNonAdmis)
```

```
admis : {'etudiant_1': 13, 'etudiant_2': 17, 'etudiant_4': 15, 'etudiant_6': 14, 'etudiant_7': 16, 'etudiant_8': 12, 'etudiant_9': 13, 'etudiant_10': 15, 'etudiant_11': 14, 'etudiant_13': 10, 'etudiant_14': 12, 'etudiant_15': 13, 'etudiant_17': 12, 'etudiant_18': 15, 'etudiant_20': 17}
non admis : {'etudiant_3': 9, 'etudiant_5': 8, 'etudiant_112': 9, 'etudiant_16': 7, 'etudiant_19': 9}
```

Corrigé du TD 11

Vous pouvez retrouver le corrigé de ce TD [ici \(Corrig%C3%A9s/Corrig%C3%A9_TD%2011.ipynb\)](#).