LES LISTES :

```
a = "Banana"
b = "Banana"

"""Imprimer les adresses mémoires de a et b"""
print(id(a))
print(id(b))
```

```
2600309128368
2600309128368
```

Dessine N caractères :

```
"""Dessiner 50 caractères au choix"""

def etoiles(car):
    print()
    print(car * 50)
    print()

etoiles("*")
etoiles("-")
```

```
**************************************************
--------------------------------------------------
```

IMPRIME LISTE :

```
a = [81,82,83]
b = [81,82,83]

print(a is b)

print(a == b)

print(id(a))
print(id(b))
```

```
False
True
2600309066176
2600307909696
```

Affectation listes :

```
alist = [4,2,8,6,5]
blist = alist
blist[3] = 999
print(alist)

alist1 = [4,2,8,6,5]
print(alist1)

blist1 = alist1 * 2
print(blist1)

blist1[3] = 999
print (blist1)
```

```
[4, 2, 8, 999, 5]

[4, 2, 8, 6, 5]

[4, 2, 8, 6, 5, 4, 2, 8, 6, 5]

[4, 2, 8, 999, 5, 4, 2, 8, 6, 5]
```

Liste slicing :

```
liste = [4,2,8,6,5]
print(liste)
del (liste[:3])
print(len(liste))

lst = ['mercury', 'venus', 'earth', 'mars', 'jupiter',
'saturn', 'uranus', 'neptune', 'pluto']
lst.remove('pluto')
first_three = lst[:3]
print(first_three)
```

```
[4, 2, 8, 6, 5]

2

['mercury', 'venus', 'earth']
```

remove :

```
b = ['q', 'u', 'i']
print(b)

z = b

print(z)
b[1] = 'i'

print(b)
print(z)

z.remove('i')
print(z)
print(b)
```

```
'q', 'u', 'i']
```

```
['q', 'u', 'i']
```

```
['q', 'i', 'i']
```

```
['q', 'i', 'i']
```

```
['q', 'i']
```

```
['q', 'i']
```

Affectation 2 :

```
sent = "Holidays can be a fun time when you have good company !"
phrase = sent
phrase = phrase + " Holidays can also be fun on your own !"
print(phrase)
```

```
Holidays can be a fun time when you have good company ! Holidays can also be fun on your own !
```

The bests occurrence characters :

```python
sally = "sally sells sea shells by the sea shore"

characters = {}

for c in sally:
    # print(c)
    if c not in characters:
        characters[c] = 0
    characters[c] = characters[c] + 1

print(characters)
list_chars_keys = list(characters.keys())
print(list_chars_keys)
best_char =  list_chars_keys[0]

for best in characters:
    if characters[best] > characters[best_char]:
        best_char = best

print("The best Character is '{}' with {}
occurrences".format(best_char,characters[best_char]))
```

```
{'s': 8, 'a': 3, 'l': 6, 'y': 2, ' ': 7, 'e': 6, 'h': 3, 'b': 1, 't': 1, 'o': 1,
'r': 1}

['s', 'a', 'l', 'y', ' ', 'e', 'h', 'b', 't', 'o', 'r']

The best Character is 's'with 8 occurrences
```

Ecrire un prg qui trouve la clé qui correspond à la valeur maximale du dictionnaire d

```
d = {'a': 194, 'b': 54, 'c':34, 'd': 44, 'e': 312, 'full':31}

ks = d.keys()
# initialize variable best_key_so_far to be the first key in d
best_key_so_far = list(ks)[0]
for k in ks:
    # check if the value associated with the current key is
    # bigger than the value associated with the best_key_so_far
    if d[k] > d[best_key_so_far]:
        # if so, save the current key as the best so far
        best_key_so_far = k



print("key " + "<" + best_key_so_far + ">" + " has the highest
value, " + str(d[best_key_so_far]))
```

```
key <e> has the highest value, 312
```

Given the dictionary swimmers, add an additional key-value pair to the dictionary with “Phelps” as the key and the integer 23 as the value. Do not rewrite the entire dictionary.

```
swimmers = {'Manuel': 4, 'Lochte': 12, 'Adrian': 7, 'Ledecky':
5, 'Dirado': 4, "Phelps": 23}

print(swimmers)
print(swimmers.values())
print(swimmers.keys())
print(swimmers.items())

print(swimmers.values()[0])
print(swimmers.keys()[0])
```

```
{'Manuel': 4, 'Lochte': 12, 'Adrian': 7, 'Ledecky': 5, 'Dirado': 4, 'Phelps': 23}

dict_values([4, 12, 7, 5, 4, 23])

dict_keys(['Manuel', 'Lochte', 'Adrian', 'Ledecky', 'Dirado', 'Phelps'])

dict_items([('Manuel', 4), ('Lochte', 12), ('Adrian', 7), ('Ledecky', 5), ('Dirado',
4), ('Phelps', 23)])
```

Add the string "hockey" as a key to the dictionary sports_periods and assign it the value of 3. Do not rewrite the entire dictionary.

```
sports_periods = {'baseball': 9, 'basketball': 4, 'soccer': 4,
'cricket': 2, "hockey": 3}

print(list(sports_periods.items()))
```

```
[('baseball', 9), ('basketball', 4), ('soccer', 4), ('cricket', 2), ('hockey', 3)]
```

The dictionary golds contains information about how many gold medals each country won in the 2016 Olympics. But today, Spain won 2 more gold medals. Update golds to reflect this information.

```
golds = {"Italy": 12, "USA": 33, "Brazil": 15, "China": 27,
"Spain": 19, "Canada": 22, "Argentina": 8, "England": 29}
golds["Spain"] = golds["Spain"] + 2

print(golds.get("Spain"))
```

```
21
```

Create a list of the countries that are in the dictionary golds, and assign that list to the variable name countries. Do not hard code this.

```
golds = {"Italy": 12, "USA": 33, "Brazil": 15, "China": 27,
"Spain": 19, "Canada": 22, "Argentina": 8, "England": 29}

countries = golds

print(countries.keys())
```

```
dict_keys(['Italy', 'USA', 'Brazil', 'China', 'Spain', 'Canada', 'Argentina',
'England'])
```

Provided is the dictionary, medal_count, which lists countries and their respective medal count at the halfway point in the 2016 Rio Olympics. Using dictionary mechanics, assign the medal count value for "Belarus" to the variable belarus. Do not hardcode this

```python
medal_count = {'United States': 70, 'Great Britain':38,
'China':45, 'Russia':30, 'Germany':17, 'Italy':22, 'France':
22, 'Japan':26, 'Australia':22, 'South Korea':14, 'Hungary':12,
'Netherlands':10, 'Spain':5, 'New Zealand':8, 'Canada':13,
'Kazakhstan':8, 'Colombia':4, 'Switzerland':5, 'Belgium':4,
'Thailand':4, 'Croatia':3, 'Iran':3, 'Jamaica':3, 'South
Africa':7, 'Sweden':6, 'Denmark':7, 'North Korea':6, 'Kenya':4,
'Brazil':7, 'Belarus':4, 'Cuba':5, 'Poland':4, 'Romania':4,
'Slovenia':3, 'Argentina':2, 'Bahrain':2, 'Slovakia':2,
'Vietnam':2, 'Czech Republic':6, 'Uzbekistan':5}
# belarus = medal_count["Belarus"]
belarus = medal_count.get("Belarus")
print(belarus)
```

```
13
```

Here's a table of English to Pirate translations :

English	Pirate
sir	matey
hotel	fleabag inn
student	swabbie
boy	matey
boy	matey
professor	foul blaggart
restaurant	galley

Write a program that asks the user for a sentence in English and then translates that sentence to Pirate.

```
pirate = {'sir': 'matey', 'hotel': 'fleabag inn', 'student':
'swabbie', 'boy': 'matey', 'restaurant': 'galley'}
#and so on

sentence = input("Please enter a sentence in English")

psentence = []
words = sentence.split()
for aword in words:
    if aword in pirate:
        psentence.append(pirate[aword])
    else:
        psentence.append(aword)

print(" ".join(psentence))
```

```
swabbie
```

Write a program that finds the most used 7 letter word in scarlet3.txt.

```
f = open('scarlet3.txt', 'r')
contents = f.read()
d = {}

for w in contents.split():
    if len(w) == 7:
        if w not in d:
            d[w] = 1
        else:
            d[w] = d[w] + 1

dkeys = d.keys()
most_used = dkeys[0]
for k in dkeys:
    if d[k] > d[most_used]:
        most_used = k

print("The most used word is '"+most_used+"', which is used
"+str(d[most_used])+" times")
```

```
Créer un fichier d'abord et ensuite tester.
```

Write a program that allows the user to enter a string. It then prints a table of the letters of the alphabet in alphabetical order which occur in the string together with the number of times each letter occurs. Case should be ignored. A sample run of the program might look this:

```python
x = input("Enter a sentence")

x = x.lower()   # convert to all lowercase

alphabet = 'abcdefghijklmnopqrstuvwxyz'

letter_count = {} # empty dictionary
for char in x:
    if char in alphabet: # ignore any punctuation, numbers, etc
        if char in letter_count:
            letter_count[char] = letter_count[char] + 1
        else:
            letter_count[char] = 1

keys = letter_count.keys()
for char in sorted(keys):
    print(char, letter_count[char])
```

```
a 1

h 1

i 1

l 1

s 1
```

Provided is a dictionary called US_medals which has the first 70 metals that the United States has won in 2016, and in which category they have won it in. Using dictionary mechanics, assign the value of the key "Fencing" to a variable fencing_value. Remember, do not hard code this.

```
US_medals = {"Swimming": 33, "Gymnastics": 6, "Track & Field":
6, "Tennis": 3, "Judo": 2, "Rowing": 2, "Shooting": 3,
             "Cycling - Road": 1, "Fencing": 70, "Diving": 2,
"Archery": 2, "Cycling - Track": 1, "Equestrian": 2,
             "Golf": 1, "Weightlifting": 1}

# fencing_value = US_medals.get("Fencing")
fencing_value = US_medals["Fencing"]
print(fencing_value)

print(list(US_medals.keys()))
print(list(US_medals.values()))
```

```
70
```

```
['Swimming', 'Gymnastics', 'Track & Field', 'Tennis', 'Judo', 'Rowing', 'Shooting',
'Cycling - Road', 'Fencing', 'Diving', 'Archery', 'Cycling - Track', 'Equestrian',
'Golf', 'Weightlifting']
```

```
[33, 6, 6, 3, 2, 2, 3, 1, 70, 2, 2, 1, 2, 1, 1]
```

The dictionary Junior shows a schedule for a junior year semester. The key is the course name and the value is the number of credits. Find the total number of credits taken this semester and assign it to the variable credits. Do not hardcode this – use dictionary accumulation !

```
Junior = {'SI 206':4, 'SI 310':4, 'BL 300':3, 'TO 313':3, 'BCOM
350':1, 'MO 300':3}

credits = 0

for j in Junior:
    credits += Junior[j]

print("The total of credits  is {}".format(credits))
```

```
The total of credits is 18
```

Create a dictionary, freq, that displays each character in string str1 as the key and its frequency as the value.

```python
str1 = "peter piper picked a peck of pickled peppers"
freq = {}

for c in str1:
    if c  not in freq:
        freq[c] = 0
    freq[c] += 1
print(freq)
```

```
{'p': 9, 'e': 8, 't': 1, 'r': 3, ' ': 7, 'i': 3, 'c': 3, 'k': 3, 'd': 2, 'a': 1,
'o': 1, 'f': 1, 'l': 1, 's': 1}
```

Provided is a string saved to the variable name s1. Create a dictionary named counts that contains each letter in s1 and the number of times it occurs.

```python
s1 = "hello"
counts = {}

for c in s1:
    if c  not in counts:
        counts[c] = 0
    counts[c] += 1
print(counts)
```

```
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

Create a dictionary, freq_words, that contains each word in string str1 as the key and its frequency as the value.

```
str1 = "I wish I wish with all my heart to fly with dragons in
a land apart"
freq_words = {}
list_str1 = list(str1.split())
print(list_str1)
for word in list_str1:
    if word  not in freq_words:
        freq_words[word] = 0
    freq_words[word] += 1
print(freq_words)
```

```
['I', 'wish', 'I', 'wish', 'with', 'all', 'my', 'heart', 'to', 'fly', 'with',
'dragons', 'in', 'a', 'land', 'apart']
```

```
{'I': 2, 'wish': 2, 'with': 2, 'all': 1, 'my': 1, 'heart': 1, 'to': 1, 'fly': 1,
'dragons': 1, 'in': 1, 'a': 1, 'land': 1, 'apart': 1}
```

Create a dictionary called wrd_d from the string sent, so that the key is a word and the value is how many times you have seen that word.

```
sent = "Singing in the rain and playing in the rain are two
entirely different situations but both can be good"
wrd_d = {}

list_sent = list(sent.split())
print(list_sent)
for word in list_sent:
    if word not in wrd_d:
        wrd_d[word] = 0
    wrd_d[word] += 1
print(wrd_d)
```

```
['Singing', 'in', 'the', 'rain', 'and', 'playing', 'in', 'the', 'rain', 'are',
'two', 'entirely', 'different', 'situations', 'but', 'both', 'can', 'be', 'good']
```

```
{'Singing': 1, 'in': 2, 'the': 2, 'rain': 2, 'and': 1, 'playing': 1, 'are': 1,
'two': 1, 'entirely': 1, 'different': 1, 'situations': 1, 'but': 1, 'both': 1,
'can': 1, 'be': 1, 'good': 1}
```

Create the dictionary characters that shows each character from the string sally and its frequency. Then, find the most frequent letter based on the dictionary. Assign this letter to the variable best_char.

```python
sally = "sally sells sea shells by the sea shore"

characters = {}

for c in sally:
    # print(c)
    if c not in characters:
        characters[c] = 0
    characters[c] = characters[c] + 1

print(characters)
list_chars_keys = list(characters.keys())
print(list_chars_keys)
best_char = list_chars_keys[0]

for best in characters:
    if characters[best] > characters[best_char]:
        best_char = best

print("The best Character is '{}' with {}
occurences".format(best_char,characters[best_char]))
```

```
{'s': 8, 'a': 3, 'l': 6, 'y': 2, ' ': 7, 'e': 6, 'h': 3, 'b': 1, 't': 1, 'o': 1,
'r': 1}
```

```
['s', 'a', 'l', 'y', ' ', 'e', 'h', 'b', 't', 'o', 'r']
```

```
The best Character is 's’with 8 occurrences
```

Find the least frequent letter. Create the dictionary characters that shows each character from string sally and its frequency. Then, find the least frequent letter in the string and assign the letter to the variable worst_char.

```
sally = "sally sells sea shells by the sea shore and by the
road"
characters = {}

for c in sally:
    # print(c)
    if c not in characters:
        characters[c] = 0
    characters[c] = characters[c] + 1
print(characters)
list_chars_keys = list(characters.keys())
print(list_chars_keys)
worst_char = list_chars_keys[0]

for best in characters:
    if characters[best] < characters[worst_char]:
        worst_char = best
print("The best Character is '{}' with {}
occurences".format(worst_char,characters[worst_char]))
```

```
{'s': 8, 'a': 5, 'l': 6, 'y': 3, ' ': 11, 'e': 7, 'h': 4, 'b': 2, 't': 2, 'o': 2,
'r': 2, 'n': 1, 'd': 2}
```

```
['s', 'a', 'l', 'y', ' ', 'e', 'h', 'b', 't', 'o', 'r', 'n', 'd']
```

```
The best Character is 'n’with 1 occurrences
```

Create a dictionary named letter_counts that contains each letter and the number of times it occurs in string1. Challenge : Letters should not be counted separately as upper-case and lower-case. Intead, all of them should be counted as lower-case.

```python
string1 = "There is a tide in the affairs of men, Which taken at the flood, leads on to fortune. Omitted, all the voyage of their life is bound in shallows and in miseries. On such a full sea are we now afloat. And we must take the current when it serves, or lose our ventures."
letter_counts = {}
string_minus = string1.lower()
print(string1)
print(string_minus)
for c in string_minus:
    if c not in letter_counts:
        letter_counts[c] = 0
    letter_counts[c] = letter_counts[c] + 1
print(letter_counts)
```

```
There is a tide in the affairs of men, Which taken at the flood, leads on to
fortune. Omitted, all the voyage of their life is bound in shallows and in miseries.
On such a full sea are we now afloat. And we must take the current when it serves,
or lose our ventures.
```

```
{'t': 19, 'h': 11, 'e': 29, 'r': 12, ' ': 53, 'i': 14, 's': 15, 'a': 17, 'd': 7,
'n': 15, 'f': 9, 'o': 17, 'm': 4, ',': 4, 'w': 6, 'c': 3, 'k': 2, 'l': 11, 'u': 8,
'.': 4, 'v': 3, 'y': 1, 'g': 1, 'b': 1}
```

Create a dictionary called low_d that keeps track of all the characters in the string p and notes how many times each character was seen. Make sure that there are no repeats of characters as keys, such that “T” and “t” are both seen as a “t” for example.

```
p = "Summer is a great time to go outside. You have to be 
careful of the sun though because of the heat."

low_d = {}
string_minus = p.lower()
print(p)
print(string_minus)

for c in string_minus:

    if c not in low_d:
        low_d[c] = 0
    low_d[c] = low_d[c] + 1

print(low_d)
```

```
Summer is a great time to go outside. You have to be careful of the sun though 
because of the heat.

{'s': 5, 'u': 7, 'm': 3, 'e': 12, 'r': 3, ' ': 20, 'i': 3, 'a': 6, 'g': 3, 't': 9, 
'o': 8, 'd': 1, '.': 2, 'y': 1, 'h': 6, 'v': 1, 'b': 2, 'c': 2, 'f': 3, 'l': 1, 'n': 
1}
```

Trouver le total de credits dans Junior Dictionary

The dictionary Junior shows a schedule for a junior year semester. The key is the course name and the value is the number of credits. Find the total number of credits taken this semester and assign it to the variable credits. Do not hardcode this - use dictionary accumulation !

```
Junior = {'SI 206':4, 'SI 310':4, 'BL 300':3, 'TO 313':3, 'BCOM 
350':1, 'MO 300':3}
credits = 0

for j in Junior:
    credits += Junior[j]

print("The total of credits  is {}".format(credits))
```

```
The total of credits is 18
```

Création d’un dictionnaire contenant les caractères de Str1 avec leurs nombres d’occurrences

Create a dictionary, freq, that displays each character in string str1 as the key and its frequency as the value.

```
str1 = "peter piper picked a peck of pickled peppers"
freq = {}

for c in str1:
    if c  not in freq:
        freq[c] = 0
    freq[c] += 1
print(freq)
```

```
{'p': 9, 'e': 8, 't': 1, 'r': 3, ' ': 7, 'i': 3, 'c': 3, 'k': 3, 'd': 2, 'a': 1,
'o': 1, 'f': 1, 'l': 1, 's': 1}
```

Création d’un dictionnaire avec le nombre d’occurrence de ses lettres

Provided is a string saved to the variable name s1. Create a dictionary named counts that contains each letter in s1 and the number of times it occurs.

```
s1 = "hello"
counts = {}

for c in s1:
    if c  not in counts:
        counts[c] = 0
    counts[c] += 1
print(counts)
```

```
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

Création d’un dictionnaire avec le nombre d’occurrence de ses phrases

Create a dictionary, freq_words, that contains each word in string str1 as the key and its frequency as the value.

```python
str1 = "I wish I wish with all my heart to fly with dragons in a land apart"
freq_words = {}
list_str1 = list(str1.split())
print(list_str1)
for word in list_str1:
    if word  not in freq_words:
        freq_words[word] = 0
    freq_words[word] += 1
print(freq_words)
```

```
['I', 'wish', 'I', 'wish', 'with', 'all', 'my', 'heart', 'to', 'fly', 'with',
'dragons', 'in', 'a', 'land', 'apart']
```

```
{'I': 2, 'wish': 2, 'with': 2, 'all': 1, 'my': 1, 'heart': 1, 'to': 1, 'fly': 1,
'dragons': 1, 'in': 1, 'a': 1, 'land': 1, 'apart': 1}
```

Création d’un dictionnaire avec le nombre d’occurrence de ses phrases

Create a dictionary called wrd_d from the string sent, so that the key is a word and the value is how many times you have seen that word.

```python
sent = "Singing in the rain and playing in the rain are two entirely different situations but both can be good"
wrd_d  = {}

list_sent = list(sent.split())
print(list_sent)
for word in list_sent:
    if word  not in wrd_d:
        wrd_d[word] = 0
    wrd_d[word] += 1
print(wrd_d)
```

```
['Singing', 'in', 'the', 'rain', 'and', 'playing', 'in', 'the', 'rain', 'are', 'two',
'entirely', 'different', 'situations', 'but', 'both', 'can', 'be', 'good']
```

```
{'Singing': 1, 'in': 2, 'the': 2, 'rain': 2, 'and': 1, 'playing': 1, 'are': 1, 'two':
1, 'entirely': 1, 'different': 1, 'situations': 1, 'but': 1, 'both': 1, 'can': 1,
'be': 1, 'good': 1}
```

Création d'un dictionnaire avec ses nombres d'occurrences de caractères, afficher la clé du caractère le plus répété (le max des occurrences)

Create the dictionary characters that shows each character from the string sally and its frequency. Then, find the most frequent letter based on the dictionary. Assign this letter to the variable best_char.

```python
sally = "sally sells sea shells by the sea shore"

characters = {}

for c in sally:
    # print(c)
    if c not in characters:
        characters[c] = 0
    characters[c] = characters[c] + 1

print(characters)
list_chars_keys = list(characters.keys())
print(list_chars_keys)
best_char =  list_chars_keys[0]

for best in characters:
    if characters[best] > characters[best_char]:
        best_char = best

print("The best Character is '{}' with {} occurrences".format(best_char,characters[best_char]))
```

```
{'s': 8, 'a': 3, 'l': 6, 'y': 2, ' ': 7, 'e': 6, 'h': 3, 'b': 1, 't': 1, 'o': 1, 'r': 1}

['s', 'a', 'l', 'y', ' ', 'e', 'h', 'b', 't', 'o', 'r']

The best Character is 's’with 8 occurrences
```

Imprimer la clé la moins répétée dans le dictionnaire

Find the least frequent letter. Create the dictionary characters that shows each character from string sally and its frequency. Then, find the least frequent letter in the string and assign the letter to the variable worst_char.

```python
sally = "sally sells sea shells by the sea shore and by the road"

characters = {}

for c in sally:
    # print(c)
    if c not in characters:
        characters[c] = 0
    characters[c] = characters[c] + 1

print(characters)
list_chars_keys = list(characters.keys())
print(list_chars_keys)
worst_char = list_chars_keys[0]

for best in characters:
    if characters[best] < characters[worst_char]:
        worst_char = best

print("The best Character is '{}' with {} occurrences".format(worst_char,characters[worst_char]))
```

```
{'s': 8, 'a': 5, 'l': 6, 'y': 3, ' ': 11, 'e': 7, 'h': 4, 'b': 2, 't': 2, 'o': 2, 'r': 2, 'n': 1, 'd': 2}

['s', 'a', 'l', 'y', ' ', 'e', 'h', 'b', 't', 'o', 'r', 'n', 'd']

The best Character is 'n'with 1 occurrences
```

Imprimer la clé la moins répétée dans le dictionnaire

Find the least frequent letter. Create the dictionary characters that shows each character from string sally and its frequency. Then, find the least frequent letter in the string and assign the letter to the variable worst_char.

```python
sally = "sally sells sea shells by the sea shore and by the road"
characters = {}
for c in sally:
    # print(c)
    if c not in characters:
        characters[c] = 0
    characters[c] = characters[c] + 1
print(characters)
list_chars_keys = list(characters.keys())
print(list_chars_keys)
worst_char = list_chars_keys[0]

for best in characters:
    if characters[best] < characters[worst_char]:
        worst_char = best
print("The best Character is '{}' with {} occurences".format(worst_char,characters[worst_char]))
```

```python
sally = "sally sells sea shells by the sea shore and by the road"
characters = {}

for c in sally:
    # print(c)
    if c not in characters:
        characters[c] = 0
    characters[c] = characters[c] + 1
print(characters)
list_chars_keys = list(characters.keys())
print(list_chars_keys)
worst_char = list_chars_keys[0]

for best in characters:
    if characters[best] < characters[worst_char]:
        worst_char = best
print("The best Character is '{}' with {} occurences".format(worst_char,characters[worst_char]))
```

```
{'s': 8, 'a': 5, 'l': 6, 'y': 3, ' ': 11, 'e': 7, 'h': 4, 'b': 2, 't': 2, 'o': 2,
'r': 2, 'n': 1, 'd': 2}
```

```
['s', 'a', 'l', 'y', ' ', 'e', 'h', 'b', 't', 'o', 'r', 'n', 'd']
```

```
The best Character is 'n'with 1 occurrences
```

Remette toutes les lettres en minuscule et construire un dictionnaire avec le nombre d'occurrence de chaque lettre

Create a dictionary named letter_counts that contains each letter and the number of times it occurs in string1. Challenge : Letters should not be counted separately as upper-case and lower-case. Intead, all of them should be counted as lower-case.

```python
string1 = "There is a tide in the affairs of men, Which taken at the flood, leads on to fortune. Omitted, all the voyage of their life is bound in shallows and in miseries. On such a full sea are we now afloat. And we must take the current when it serves, or lose our ventures."
letter_counts = {}
string_minus = string1.lower()
print(string1)
print(string_minus)
for c in string_minus:

    if c not in letter_counts:
        letter_counts[c] = 0
    letter_counts[c] = letter_counts[c] + 1

print(letter_counts)
```

```
There is a tide in the affairs of men, Which taken at the flood, leads on to
fortune. Omitted, all the voyage of their life is bound in shallows and in miseries.
On such a full sea are we now afloat. And we must take the current when it serves,
or lose our ventures.
```

```
{'t': 19, 'h': 11, 'e': 29, 'r': 12, ' ': 53, 'i': 14, 's': 15, 'a': 17, 'd': 7,
'n': 15, 'f': 9, 'o': 17, 'm': 4, ',': 4, 'w': 6, 'c': 3, 'k': 2, 'l': 11, 'u': 8,
'.': 4, 'v': 3, 'y': 1, 'g': 1, 'b': 1}
```

Remette toutes les lettres en minuscule et construire un dictionnaire avec le nombre d’occurrence de chaque lettre. NB : T et t est un seul caractère, pas de différence entre majuscule et minuscule.

Create a dictionary called low_d that keeps track of all the characters in the string p and notes how many times each character was seen. Make sure that there are no repeats of characters as keys, such that “T” and “t” are both seen as a “t” for example.

```python
p = "Summer is a great time to go outside. You have to be careful of the sun though because of the heat."

low_d = {}
string_minus = p.lower()
print(p)
print(string_minus)

for c in string_minus:

    if c not in low_d:
        low_d[c] = 0
    low_d[c] = low_d[c] + 1

print(low_d)
```

```
Summer is a great time to go outside. You have to be careful of the sun though
because of the heat.

{'s': 5, 'u': 7, 'm': 3, 'e': 12, 'r': 3, ' ': 20, 'i': 3, 'a': 6, 'g': 3, 't': 9,
'o': 8, 'd': 1, '.': 2, 'y': 1, 'h': 6, 'v': 1, 'b': 2, 'c': 2, 'f': 3, 'l': 1, 'n':
1}
```

Les Tuples

Provided is a list of tuples. Create another list called t_check that contains the third element of every tuple.

```
lst_tups = [('Articuno', 'Moltres', 'Zaptos'), ('Beedrill',
'Metapod', 'Charizard', 'Venasaur', 'Squirtle'), ('Oddish',
'Poliwag', 'Diglett', 'Bellsprout'), ('Ponyta', "Farfetch'd",
"Tauros", 'Dragonite'), ('Hoothoot', 'Chikorita', 'Lanturn',
'Flaaffy', 'Unown', 'Teddiursa', 'Phanpy'), ('Loudred',
'Volbeat', 'Wailord', 'Seviper', 'Sealeo')]

t_check = []

for item in lst_tups:
    t_check.append(item[2])

print(t_check)
```

```
['Zaptos', 'Charizard', 'Diglett', 'Tauros', 'Lanturn', 'Wailord']
```

Below, we have provided a list of tuples. Write a for loop that saves the second element of each tuple into a list called seconds.

```
tups = [('a', 'b', 'c'), (8, 7, 6, 5), ('blue', 'green',
'yellow', 'orange', 'red'), (5.6, 9.99, 2.5, 8.2), ('squirrel',
'chipmunk')]

seconds = []

for item in tups:
    seconds.append(item[1])

print(seconds)
```

```
['b', 7, 'green', 9.99, 'chipmunk']
```

Define a function called information that takes as input, the variables name, birth_year, fav_color, and hometown. It should return a tuple of these variables in this order.

```
def information(name,birth_day,fav_color,home_town):
    return name, birth_day, fav_color, home_town

print(information("salih","26/12/1965","Green","Sétif"))
```

```
('salih', '26/12/1965', 'Green', 'Sétif')
```

Deuxième version de l'exercice qui précède :

```
def information(name, birth_day, fav_color, home_town):
    return name, birth_day, fav_color, home_town

nom , date_naissance, couleur_fav, ville =
information("salih","26/12/1965","Green","Sétif")

print(nom , date_naissance, couleur_fav, ville)
```

```
salih 26/12/1965 Green Sétif
```

Define a function called info with the following required parameters : name, age, birth_year, year_in_college, and hometown. The function should return a tuple that contains all the inputted information.

```
def info(name,âge,birth_year,year_in_college,hometown):
    return name, âge, birth_year, year_in_college, hometown

print(info("Salih","56","1965","5","Sétif"))
```

```
('Salih', '56', '1965', '5', 'Sétif')
```

Deuxième méthode de l'exercice au-dessus :

```
def information(name,birth_day,fav_color,home_town):
    return name, birth_day, fav_color, home_town
nom , date_naissance, couleur_fav, ville =
information("salih","26/12/1965","Green","Sétif")

print(nom , date_naissance, couleur_fav, ville)
def information(name,birth_day,fav_color,home_town):
    return name, birth_day, fav_color, home_town

nom , date_naissance, couleur_fav, ville =
information("salih","26/12/1965","Green","Sétif")

print(nom , date_naissance, couleur_fav, ville)
```

```
salih 26/12/1965 Green Sétif
```

```
 Define a function called info with the following required
parameters : name, age, birth_year, year_in_college, and
hometown. The function should return a tuple that contains all
the inputted information.

def info(name,âge,birth_year,year_in_college,hometown):
    return name, âge, birth_year, year_in_college, hometown

print(info("Salih","56","1965","5","Sétif"))
```

```
('Salih', '56', '1965', '5', 'Sétif')
```

Deuxième méthode de l'exercice au-dessus :

```
def info(name,âge,birth_year,year_in_college,hometown):
    return name, âge, birth_year, year_in_college, hometown

nom,age,annee_naiss,annees_college,ville =
info("Salih","56","1965","5","Sétif")
print(nom,age,annee_naiss,annees_college,ville)
```

```
Salih 56 1965 5 Sétif
```

Tuple Unpacking :

Swapping Values between Variables with unpacking tuple

```
a = 1
b = 2
(a, b) = (b, a)
print(a, b)
```

```
2 1
```

Unpacking Into Iterator Variables

```
authors = [('Paul', 'Resnick'), ('Brad', 'Miller'), ('Lauren', 
'Murphy')]
for first_name, last_name in authors:
    print("first name:", first_name, "last name:", last_name)
```

```
first name: Paul last name: Resnick

first name: Brad last name: Miller

first name: Lauren last name: Murphy
```

The Pythonic Way to Enumerate Items in a Sequence

```
fruits = ['apple', 'pear', 'apricot', 'cherry', 'peach']
for n in range(len(fruits)):
    print(n, fruits[n])
```

```
0 apple

1 pear

2 apricot

3 cherry

4 peach
```

We are now prepared to understand a more pythonic approach to enumerating items in a sequence. Python provides a built-in function enumerate. It takes a sequence as input and returns a sequence of tuples. In each tuple, the first element is an integer and the second is an item from the original sequence. (It actually produces an "iterable" rather than a list, but we can use it in a for loop as the sequence to iterate over.)

```python
fruits = ['apple', 'pear', 'apricot', 'cherry', 'peach']
for item in enumerate(fruits):
    print(item[0], item[1])
```

```
0 apple

1 pear

2 apricot

3 cherry

4 peach
```

If you remember, the .items() dictionary method produces a sequence of tuples. Keeping this in mind, we have provided you a dictionary called pokemon. For every key value pair, append the key to the list p_names, and append the value to the list p_number. Do not use the .keys() or .values() methods.

```python
pokemon = {'Rattata': 19, 'Machop': 66, 'Seel': 86, 'Volbeat':
86, 'Solrock': 126}

p_number = []
p_names = []

for k,v in pokemon.items():
    p_number.append(v)
    p_names.append(k)

print(p_names)
print(p_number)
```

```
['Rattata', 'Machop', 'Seel', 'Volbeat', 'Solrock']

[19, 66, 86, 86, 126]
```

The .items() method produces a sequence of key-value pair tuples. With this in mind, write code to create a list of keys from the dictionary track_medal_counts and assign the list to the variable name track_events. Do NOT use the .keys() method.

```
track_medal_counts = {'shot put': 1, 'long jump': 3, '100
meters': 2, '400 meters': 2, '100 meter hurdles': 3, 'triple
jump': 3, 'steeplechase': 2, '1500 meters': 1, '5K': 0, '10K':
0, 'marathon': 0, '200 meters': 0, '400 meter hurdles': 0,
'high jump': 1}

track_events = []

for k,v in track_medal_counts.items():
    track_events.append(k)

print("The list of keys of dictionnary is :
{}".format(track_events))
```

```
The list of keys of dictionnary is : ['shot put', 'long jump', '100 meters', '400
meters', '100 meter hurdles', 'triple jump', 'steeplechase', '1500 meters', '5K',
'10K', 'marathon', '200 meters', '400 meter hurdles', 'high jump']
```

Create a tuple called olympics with four elements : “Beijing”, “London”, “Rio”, “Tokyo”.

```
olympics = ("Beijing", "London", "Rio", "Tokyo")
print(olympics)
```

```
('Beijing', 'London', 'Rio', 'Tokyo')
```

The list below, tuples_lst, is a list of tuples. Create a list of the second elements of each tuple and assign this list to the variable country.

```
tuples_lst = [('Beijing', 'China', 2008), ('London', 'England',
2012), ('Rio', 'Brazil', 2016, 'Current'), ('Tokyo', 'Japan',
2020, 'Future')]
country = []

for item in tuples_lst:
    country.append(item)

print("The list of second element of each tuple is :
{}".format(country))
```

```
The list of second element of each tuple is : [('Beijing', 'China', 2008),
('London', 'England', 2012), ('Rio', 'Brazil', 2016, 'Current'), ('Tokyo', 'Japan',
2020, 'Future')]
```

With only one line of code, assign the variables city, country, and year to the values of the tuple olymp.

```
olymp = ('Rio', 'Brazil', 2016)

city, country, year = olymp

print(city,country,year)

print("The city is {}, country is {} and year is
{}".format(city,country,year))
```

```
Rio Brazil 2016

The city is Rio, country is Brazil and year is 2016
```

Define a function called info with five parameters: name, gender, age, bday_month, and hometown. The function should then return a tuple with all five parameters in that order.

```
def info(name,gender,âge,bday_month,hometown):
    return name,gender,âge,bday_month,hometown

print(info("salih","Mister","56","december","Constantine"))
```

```
('salih', 'Mister', '56', 'december', 'Constantine')
```

Given is the dictionary, gold, which shows the country and the number of gold medals they have earned so far in the 2016 Olympics. Create a list, num_medals, that contains only the number of medals for each country. You must use the .items() method. Note: The .items() method provides a list of tuples. Do not use .keys() method.

```python
gold = {'USA':31, 'Great Britain':19, 'China':19, 'Germany':13,
'Russia':12, 'Japan':10, 'France':8, 'Italy':8}

num_medals = []

for item in gold.items():
    num_medals.append(item[1])

print("The list of medals is  : {}".format(num_medals))
```

```
The list of medals is : [31, 19, 19, 13, 12, 10, 8, 8]
```

Use a for loop to print out the last name, year of birth, and city for each of the people. (There are multiple ways you could do this. Try out some code and see what happens !)

```python
julia = ("Julia", "Roberts", 1967, "Duplicity", 2009,
"Actress", "Atlanta, Georgia")
claude = ("Claude", "Shannon", 1916, "A Mathematical Theory of
Communication", 1948, "Mathematician", "Petoskey, Michigan")
alan = ("Alan", "Turing", 1912, "Computing machinery and
intelligence", 1950, "Mathematician", "London, England")

"""la liste people contient les 3 tuples : julia, claude, et
alan"""
people = [julia, claude, alan]
for item in people:
    print(item[1] , item[2] , item[-1])
```

```
Roberts 1967 Atlanta, Georgia

Shannon 1916 Petoskey, Michigan

Turing 1912 London, England
```