1. قم بعد عدد مرات تكرار كل حرف في النص التالي (لا تقم بعد أي شيء أخر غير الحروف) باستخدام if

Count the number of each letter (only letters) occurrence in the following text using if statement.

```
txt = """One of the most effective ways to reduce the friction
associated with
your habits is to practice environment design
In a previous chapter we discussed environment design as a
method for making cues more
obvious but you can also optimize your environment to make
actions
easier
For example when deciding where to practice a new habit it is
best to choose a place that is already along the path of your
daily
routine
Habits are easier to build when they fit into the flow of your
life
You are more likely to go to the gym if it is on your way to
work
because stopping does not add much friction to your lifestyle
"""
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# make the string lower case and remove the new line and spaces

# create an empty dictionary to store the letters

# loop through the string

# check if the letter is not in the dictionary

# if the letter is not in the dictionary add it and set the value to 1

# if the letter is in the dictionary add 1 to the value

# print the dictionary
```

```
# make the string lower case and remove the new line and spaces
txt = txt.lower().replace("\n","").replace(" ","")


# create an empty dictionary to store the letters
letter_count = {}

# loop through the string
for letter in txt:
    # check if the letter is not in the dictionary
    if letter not in letter_count:
        # if the letter is not in the dictionary add it and set
the value to 1
        letter_count[letter] = 1

    else:
        # if the letter is in the dictionary add 1 to the value
        letter_count[letter] += 1

# print the dictionary
for letter, count in letter_count.items():
    print(f"{letter.title()}: {count}")
```

{//}

2.قم بعد عدد مرات تكرار كل حرف في النص التالي (لا تقم بعد أي شيء أخر غير الحروف) باستخدام get method

Count the number of each letter (only letters) occurrence in the following text using get method.

```
txt = """One of the most effective ways to reduce the friction
associated with
your habits is to practice environment design
In a previous chapter we discussed environment design as a
method for making cues more
obvious but you can also optimize your environment to make
actions
easier
For example when deciding where to practice a new habit it is
best to choose a place that is already along the path of your
daily
routine
Habits are easier to build when they fit into the flow of your
life
You are more likely to go to the gym if it is on your way to
work
because stopping does not add much friction to your lifestyle
"""
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# make the string lower case and remove the new line and spaces

# create an empty dictionary to store the letters

# get the value and add 1 to it
```

```
# make the string lower case and remove the new line and spaces
txt = txt.lower().replace("\n","").replace(" ","")

# create an empty dictionary to store the letters
letter_count = {}

for letter in txt:
    # get the value and add 1 to it
    letter_count[letter] = letter_count.get(letter, 0) + 1

# print the dictionary
for letter, count in letter_count.items():
    print(f"{letter.title()}: {count}")
```

{//}

3.قم بعد عدد مرات تكرار كل حرف في النص التالي (لا تقم بعد أي شيء أخر غير الحروف) باستخدام setdefault method

Count the number of each letter (only letters) occurrence in the following text using setdefault method.

```
txt = """One of the most effective ways to reduce the friction
associated with
your habits is to practice environment design
In a previous chapter we discussed environment design as a
method for making cues more
obvious but you can also optimize your environment to make
actions
easier
For example when deciding where to practice a new habit it is
best to choose a place that is already along the path of your
daily
routine
Habits are easier to build when they fit into the flow of your
life
You are more likely to go to the gym if it is on your way to
work
because stopping does not add much friction to your lifestyle
"""
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# make the string lower case and remove the new line and spaces

# create an empty dictionary to store the letters

# set the value of the letter to 0 if it is not in the
dictionary

# add 1 to the value of the letter
```

```
# make the string lower case and remove the new line and spaces
txt = txt.lower().replace("\n","").replace(" ","")

# create an empty dictionary to store the letters
letter_count = {}

for letter in txt:
    # set the value of the letter to 0 if it is not in the
dictionary
    letter_count.setdefault(letter, 0)

    # add 1 to the value of the letter
    letter_count[letter] += 1

# print the dictionary
for letter, count in letter_count.items():
    print(f"{letter.title()}: {count}")
```

4.قم بعد عدد مرات تكرار الكلمات داخل النص التالي باستخدام طريقتين على الأقل

Count the number of occurrences of each word in the following text using at least two methods.

```
txt = """One of the most effective ways to reduce the friction
associated with
your habits is to practice environment design
In a previous chapter we discussed environment design as a
method for making cues more
obvious but you can also optimize your environment to make
actions
easier
For example when deciding where to practice a new habit it is
best to choose a place that is already along the path of your
daily
routine
Habits are easier to build when they fit into the flow of your
life
You are more likely to go to the gym if it is on your way to
work
because stopping does not add much friction to your lifestyle
"""
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# make the string lower case

# split the string into a list of words

# ### method 1 ####

# create an empty dictionary to store the words

# loop through the list of words

# get the value of the word from the dictionary

# if the word is not in the dictionary set the value to 0 then add 1
#################
# ### method 2 ####

# create an empty dictionary to store the words

# loop through the list of words

# check if the word is not in the dictionary

# if the word is not in the dictionary add it and set the value to 1

# if the word is in the dictionary add 1 to the value

# print the dictionary
```

{//}

```
# make the string lower case
txt = txt.lower()

# split the string into a list of words
words = txt.split()

# ### method 1 ####
# create an empty dictionary to store the words
word_count = {}


# loop through the list of words
for word in words:
    # get the value of the word from the dictionary
    # if the word is not in the dictionary set the value to 0
then add 1
    word_count[word] = word_count.get(word, 0) + 1

print(word_count)
################
# ### method 2 ####

# create an empty dictionary to store the words
word_count = {}


# loop through the list of words
for word in words:
    # check if the word is not in the dictionary
    if word not in word_count:
        # if the word is not in the dictionary add it and set
the value to 1
        word_count[word] = 1
```

{//}

```
    else:
        # if the word is in the dictionary add 1 to the value
        word_count[word] += 1

# print the dictionary
print(word_count)

##############################
# ### method 3 ####

# create an empty dictionary to store the words
word_count = {}


# loop through the list of words
for word in words:
    # set the value of the word to 0 if it is not in the
dictionary
    word_count.setdefault(word, 0)

    # add 1 to the value of the word
    word_count[word] += 1

print(word_count)
```

5. القائمة التالية تحتوي على المنتجات التي اشتراها صديقنا سالم خلال الأيام المختلفة هذا الشهر، وكان من عادة صديقنا أنه يكتب أمام كل يوم المنتجات التي اشتراها، وإذا قام بشراء المنتج أكثر من مرة فإنه يقوم بكتابة مرة أخرى في هذه القائمة، والتي صممها لكي يتتبع على ماذا ينفق أمواله، ولكن بعد انقضاء مدة، يصبح تحديد ما هي المنتجات التي أنفق عليها صعبا كما أمامنا الآن في هذه القائمة، ولذلك سوف نقوم بمساعدته في حل هذه المشكلة، واخباره كم مرة اشترى كل منتج من المنتجات الموجودة في القائمة التي أمامنا هنا

Help our friend Salem know how many times he had bought each item in the following list.

```
february_shopping_list = {
    1: ['meat', 'chicken', 'chicken',
          'chicken', 'bread', 'chocolate', 'meat'],
    2: ['bread', 'milks', 'butter', 'butter', 'chocolate'],
    3: ['butter', 'meat', 'milks'],
    4: ['butter', 'bread', 'nuts'],
    5: ['butter', 'bread', 'chocolate', 'chocolate'],
    6: ['nuts', 'butter', 'butter',
        'butter', 'chocolate', 'butter'],
    7: ['cheese', 'milks', 'butter', 'nuts'],
    8: ['cheese', 'meat', 'nuts', 'yoghurt', 'cheese'],
    9: ['chocolate', 'milks', 'milks',
         'chocolate', 'milks', 'eggs', 'meat'],
    10: ['yoghurt', 'butter', 'chocolate', 'cheese', 'butter'],
    11: ['cheese', 'meat', 'yoghurt'],
    12: ['eggs', 'chocolate', 'meat', 'eggs', 'butter'],
    13: ['bread', 'eggs', 'yoghurt',
          'yoghurt', 'chicken', 'chocolate'],
    14: ['milks', 'meat', 'meat'],
    15: ['meat', 'chicken', 'butter', 'nuts', 'nuts'],
    16: ['meat', 'meat', 'chicken']
    }
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# create an empty dictionary to store the items and their count

# loop through the dictionary

# loop through the list of items

# get the value and add 1 to it

# print the dictionary
```

```
# create an empty dictionary to store the items and their count
item_count = {}

# loop through the dictionary
for day, items in february_shopping_list.items():
    # loop through the list of items
    for item in items:
        # get the value and add 1 to it
        item_count[item] = item_count.get(item, 0) + 1

# print the dictionary
for item, count in item_count.items():
    print(f'{item.title()}: {count}')
```

6. الآن قد عرفنا كم مرة اشترى صديقنا سالم كل منتج من المنتجات التي في القائمة، الآن جاء وقت معرفة السؤال الأهم وهو ما مقدار المال الذي أنفقه على كل منتج من المنتجات التالية، وما هو إجمالي ما أنفقه على جميع المنتجات، ولإجابة ذلك ستجد هنا بالأسفل قائمة تحتوي على السعر الخاص بكل منتج كما يلي

Now we have to answer the important question, how much did Salem spend on each item, and overall how much did he pay for all items? And to answer this here are a dictionary with the price of each item.

```
items_prices = {
    'meat': 250,
    'chicken': 140,
    'bread': 10,
    'chocolate': 20,
    'milks': 42,
    'butter': 75,
    'nuts': 90,
    'cheese': 65,
    'yoghurt': 25,
    'eggs': 120
}
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# create an empty dictionary to store the items and their count

# loop through the dictionary

# loop through the list of items

# get the value and add 1 to it

# create an empty dictionary to store the items and their cost

# loop through the dictionary

# add the item and its cost to the dictionary

# print the total cost of the items

# print the total cost of the items
```

```python
# create an empty dictionary to store the items and their count
items_count = {}

# loop through the dictionary
for day, items in february_shopping_list.items():
    # loop through the list of items
    for item in items:
        # get the value and add 1 to it
        items_count[item] = items_count.get(item, 0) + 1


# create an empty dictionary to store the items and their cost
item_cost = {}

# loop through the dictionary
for item, count in items_count.items():
    # add the item and its cost to the dictionary
    item_cost[item] = count * items_prices[item]

# print the total cost of the items
for item, cost in item_cost.items():
    print(f'{item.title()}: {cost:,} EGP')

# print the total cost of the items
total_cost = sum(item_cost.values())
print(f'The Total Cost: {total_cost:,} EGP')
```