

Frédéric  
LURET



# Python

## Banque de questions

Version du 9 décembre 2024



## 1 Site 1



En rapport avec les séries 100+  
lien vers le site d'origine 1  
lien vers le site d'origine 2

### Question 1



Écrivez un programme qui trouve tous les nombres multiples de 7 mais pas de 5, entre 2000 et 3200 (les deux inclus). Les nombres obtenus doivent être imprimés dans une séquence séparée par des virgules sur une seule ligne.

**Indices :** Utilisez la méthode range(début, fin)

**Code python :**

```
1 result = []
2 for i in range(2000, 3201):
3     if (i % 7 == 0) and (i % 5 != 0):
4         result.append(str(i))
5
6 print(",".join(result))
```

q001.py

**Code python :**

```
1 print(*(i for i in range(2000, 3201) if i%7 == 0 and i%5 != 0),
↪ sep=",")
```

q001-01.py

### Question 2



Écrivez un programme qui peut calculer la factorielle d'un nombre donné.

Supposons que l'entrée suivante soit fournie au programme :

8

Ensuite, la sortie doit être :

40320

**Code python :**



```
1 def fact(x):
2     if x == 0:
3         return 1
4     return x * fact(x - 1)
5
6 x=int(input())
7 print(fact(x))
```

q002.py

Code python :

```
1 n = int(input()) # input() function takes input as string type
2 # int() converts it to integer type
3 fact = 1
4 i = 1
5 while i <= n:
6     fact = fact * i
7     i = i + 1
8 print(fact)
```

q002-01.py

Code python :

```
1 n = int(input()) # input() function takes input as string type
2 # int() converts it to integer type
3 fact = 1
4 for i in range(1, n + 1):
5     fact = fact * i
6 print(fact)
```

q002-02.py

Code python :

```
1 n = int(input())
2
3
4 def shortFact(x):
5     return 1 if x <= 1 else x * shortFact(x - 1)
6
7
8 print(shortFact(n))
```

q002-03.py

### Question 3

Avec un nombre entier **n** donné, écrivez un programme pour générer un dictionnaire qui contient (**i, i\*i**) tel que **i** est un nombre entier entre **1** et **n** (les deux inclus). et ensuite le programme doit imprimer le dictionnaire.



Supposons que l'entrée suivante soit fournie au programme :

8

La sortie devrait alors être :

{1 : 1, 2 : 4, 3 : 9, 4 : 16, 5 : 25, 6 : 36, 7 : 49, 8 : 64}

**Code python :**

```
1 n = int(input())
2 d = {}
3 for i in range(1, n+1):
4     d[i] = i * i
5
6 print(d)
```

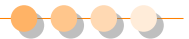
q003.py

**Code python :**

```
1 n = int(input())
2 ans = {i: i * i for i in range(1, n + 1)}
3 print(ans)
```

q003-01.py

#### Question 4



Écrire un programme qui accepte une séquence de nombres séparés par des virgules à partir de la console et qui génère une liste et un tuple contenant chaque nombre.

Supposons que l'entrée suivante soit fournie au programme :

34,67,55,33,12,98

Ensuite, la sortie doit être :

['34', '67', '55', '33', '12', '98']

('34', '67', '55', '33', '12', '98')

**Code python :**

```
1 values = input()
2 l = values.split(",")
3 t = tuple(l)
4 print(l)
5 print(t)
```

q004.py

**Code python :**



```
1 lst = input().split(",")
2 # the input is being taken as string and as it is string it has a built
  ↳ in
3 # method name split. ',' inside split function does split where it
  ↳ finds any ','
4 # and save the input as list in lst variable
5
6 tpl = tuple(lst) # tuple method converts list to tuple
7
8 print(lst)
9 print(tpl)
```

q004-01.py

### Question 5

Définir une classe qui possède au moins deux méthodes :

**getString** : pour obtenir une chaîne de caractères à partir de l'entrée de la console

**printString** : pour imprimer la chaîne en majuscules.

Veuillez également inclure une fonction de test simple pour tester les méthodes de la classe.

**Indices** : Utilisez la méthode `__init__` pour construire certains paramètres

**Code python :**

```
1 class InputOutString(object):
2     def __init__(self):
3         self.s = ""
4
5     def getString(self):
6         self.s = input()
7
8     def printString(self):
9         print(self.s.upper())
10
11 strObj = InputOutString()
12 strObj.getString()
13 strObj.printString()
```

q005.py

### Question 6

Écrivez un programme qui calcule et imprime la valeur selon la formule donnée :

$Q = \text{Racine carrée de } [(2 * C * D)/H]$

Voici les valeurs fixes de C et H :

C est 50. H est égal à 30.

D est la variable dont les valeurs doivent être introduites dans votre programme dans



une séquence séparée par des virgules.

Exemple

Supposons que le programme reçoive la séquence d'entrée suivante, séparée par des virgules :

100,150,180

La sortie du programme devrait être :

18,22,24

**Indices :** Si la sortie reçue est sous forme décimale, elle doit être arrondi à sa valeur la plus proche (par exemple, si la sortie reçue est de 26,0, elle doit être imprimée comme 26)

**Code python :**

```
1 import math
2 c = 50
3 h = 30
4 value = []
5 items = [x for x in input().split(',')]
6 for d in items:
7     value.append(str(int(round(math.sqrt(2*c*float(d)/h))))))
8
9 print(','.join(value))
```

q006.py

**Code python :**



```
1  from math import sqrt  # import specific functions as importing all
   ↪  using *
2
3  # is bad practice
4
5  C, H = 50, 30
6
7
8  def calc(D):
9      return sqrt((2 * C * D) / H)
10
11
12 D = [int(i) for i in input().split(",")]  # splits in comma position
   ↪  and set up in list
13 D = [int(i) for i in D]  # converts string to integer
14 D = [calc(i) for i in D]  # returns floating value by calc method for
   ↪  every item in D
15 D = [round(i) for i in D]  # All the floating values are rounded
16 D = [
17     str(i) for i in D
18 ]  # All the integers are converted to string to be able to apply join
   ↪  operation
19
20 print(",".join(D))
```

q006-01.py

**Code python :**

```
1  from math import sqrt
2
3  C, H = 50, 30
4
5
6  def calc(D):
7      return sqrt((2 * C * D) / H)
8
9
10 D = input().split(",")  # splits in comma position and set up in list
11 D = [
12     str(round(calc(int(i)))) for i in D
13 ]  # using comprehension method. It works in order of the previous code
14 print(",".join(D))
```

q006-02.py

**Code python :**



```
1 from math import sqrt
2
3 C, H = 50, 30
4
5
6 def calc(D):
7     return sqrt((2 * C * D) / H)
8
9
10 print(",".join([str(int(calc(int(i)))) for i in input().split(",")]))
```

q006-03.py

Code python :

```
1 from math import * # importing all math functions
2
3 C, H = 50, 30
4
5
6 def calc(D):
7     D = int(D)
8     return str(int(sqrt((2 * C * D) / H)))
9
10
11 D = input().split(",")
12 D = list(map(calc, D)) # applying calc function on D and storing as a
    ↪ list
13 print(",".join(D))
```

q006-04.py

### Question 7

Écrivez un programme qui prend 2 chiffres, X,Y en entrée et génère un tableau à 2 dimensions. La valeur de l'élément dans la i-ième ligne et la j-ième colonne du tableau doit être  $i*j$ .

Remarque :  $i = 0,1,.., X-1$  ;  $j = 0,1,..,Y-1$ .

Exemple

Supposons que les entrées suivantes soient données au programme :

3,5

La sortie du programme devrait alors être la suivante :

`[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]`

Puis un affichage sous la forme d'un tableau :

0 0 0 0 0

0 1 2 3 4

0 2 4 6 8





Code python :

```
1 input_str = input()
2 rowNum, colNum = [int(x) for x in input_str.split(',')]
3
4 multilist = []
5
6 for row in range(rowNum):
7     row_list = []
8     for col in range(colNum):
9         row_list.append(row * col)
10    multilist.append(row_list)
11
12 print(multilist)
13 print()
14 for row in multilist:
15     print(' '.join(map(str, row)))
```

q007.py

Code python :

```
1 input_str = input()
2 dimensions = [int(x) for x in input_str.split(',')]
3 rowNum = dimensions[0]
4 colNum = dimensions[1]
5
6 multilist = [[row * col for col in range(colNum)] for row in
↵   range(rowNum)]
7
8 print(multilist)
9 print()
10 for row in multilist:
11     print(' '.join(map(str, row)))
```

q007-01.py

Code python :

```
1 x, y = map(int, input().split(", "))
2 lst = []
3
4 for i in range(x):
5     tmp = []
6     for j in range(y):
7         tmp.append(i * j)
8     lst.append(tmp)
9
10 print(lst)
```

q007-02.py



Code python :

```
1 x, y = map(int, input().split(","))
2 lst = [[i * j for j in range(y)] for i in range(x)]
3 print(lst)
```

q007-03.py

### Question 8

Écrivez un programme qui accepte une séquence de mots séparée par des virgules en entrée et imprime les mots dans une séquence séparée par des virgules après les avoir triés de manière alphabétique.

Supposons que l'entrée suivante soit fournie au programme :

sans, bonjour, sac, monde

Ensuite, la sortie doit être :

Sac, bonjour, sans, monde

Code python :

```
1 lst = input().split(",")
2 lst.sort()
3 print(",".join(lst))
```

q008.py

### Question 9

Écrivez un programme qui accepte une séquence de lignes en entrée et imprime les lignes après avoir mis en majuscules tous les caractères de la phrase. La saisie d'une ligne vide lance votre traitement.

Supposons que l'entrée suivante soit fournie au programme :

Bonjour le monde

C'est en forgeant qu'on devient forgeron

La sortie devrait alors être :

BONJOUR AU MONDE

C'EST EN FORGEANT QU'ON DEVIENT FORGERON

Code python :



```
1 lines = []
2 while True:
3     s = input()
4     if s:
5         lines.append(s.upper())
6     else:
7         break
8
9 for sentence in lines:
10    print(sentence)
```

q009.py

Code python :

```
1 def user_input():
2     while True:
3         s = input()
4         if not s:
5             return
6         yield s
7
8
9 for line in map(str.upper, user_input()):
10    print(line)
```

q009-01.py

### Question 10

Écrivez un programme qui accepte une séquence de mots séparés dans l'espace en entrée et imprime les mots après avoir retiré tous les mots en double et les tris de manière alphanumérique.

Supposons que l'entrée suivante soit fournie au programme :

Bonjour le monde et la pratique rend à nouveau le monde parfait et bonjour

La sortie doit être :

Bonjour bonjour et la le monde nouveau parfait pratique rend à

**Indices :** Nous utilisons le conteneur **set** pour supprimer automatiquement les données dupliqués.

Code python :



```
1 word = input().split()
2
3 for i in word:
4     if (
5         word.count(i) > 1
6     ): # count function returns total repeation of an element that
        ↪ is send as argument
7         word.remove(i) # removes exactly one element per call
8
9 word.sort()
10 print(" ".join(word))
```

q010.py

**Code python :**

```
1 word = input().split()
2 [word.remove(i) for i in word if word.count(i) > 1] # removal
    ↪ operation with comprehension method
3 word.sort()
4 print(" ".join(word))
```

q010-01.py

**Code python :**

```
1 word = sorted(
2     list(set(input().split())))
3 ) # input string splits -> converting into set() to store unique
4 # element -> converting into list to be able to apply sort
5 print(" ".join(word))
```

q010-02.py

**Question 11**

Écrivez un programme qui accepte une séquence de nombres binaires à 4 chiffres séparés par des virgules comme entrée, puis vérifiez s'ils sont divisibles par 5 ou non. Les nombres divisibles par 5 doivent être imprimés dans une séquence séparée par des virgules.

Exemple :

0100,0011,1010,1001

Qui correspondent respectivement à 4, 3, 10 et 9.

Alors la sortie doit être :

1010

**Code python :**



```
1 value = []
2 items = [x for x in input().split(',')]
3 for p in items:
4     intp = int(p, 2)
5     print(intp)
6     if not intp % 5:
7         value.append(p)
8
9 print(', '.join(value))
```

q011.py

Code python :

```
1 def check(x): # converts binary to integer & returns zero if divisible
    ↪ by 5
2     total, pw = 0, 1
3     reversed(x)
4
5     for i in x:
6         total += pw * (ord(i) - 48) # ord() function returns ASCII
        ↪ value
7         pw *= 2
8     return total % 5
9
10
11 data = input().split(",") # inputs taken here and splited in ','
    ↪ position
12 lst = []
13
14 for i in data:
15     if check(i) == 0: # if zero found it means divisible by zero and
        ↪ added to the list
16         lst.append(i)
17
18 print(", ".join(lst))
```

q011-01.py

Code python :



```
1 def check(x): # check function returns true if divisible by 5
2     return int(x, 2) % 5 == 0 # int(x,b) takes x as string and b as
    ↪ base from which
3     # it will be converted to decimal
4
5
6 data = input().split(",")
7
8 data = list(
9     filter(check, data)
10 ) # in filter(func,object) function, elements are picked from 'data'
    ↪ if found True by 'check' function
11 print(",".join(data))
```

q011-02.py

Code python :

```
1 data = input().split(",")
2 data = list(
3     filter(lambda i: int(i, 2) % 5 == 0, data)
4 ) # lambda is an operator that helps to write function of one line
5 print(",".join(data))
```

q011-03.py

### Question 12

Écrivez un programme, qui trouvera tous les chiffres entre 1000 et 3000 (tous deux inclus) pour lesquels chaque chiffre du nombre est pair. Les nombres obtenus doivent être imprimés dans une séquence séparée par des virgules sur une seule ligne.

Code python :

```
1 values = []
2 test = False
3 for i in range(1000, 3001):
4     s = str(i)
5     test = int(s[0]) % 2 == 0 and int(s[1]) % 2 == 0
6     test = test and int(s[2]) % 2 == 0 and int(s[3]) % 2 == 0
7     if test:
8         values.append(s)
9 print(",".join(values))
```

q012.py

Code python :



```
1 lst = []
2
3 for i in range(1000, 3001):
4     flag = 1
5     for j in str(i): # every integer number i is converted into string
6         if ord(j) % 2 != 0: # ord returns ASCII value and j is every
            ↪ digit of i
7             flag = 0 # flag becomes zero if any odd digit found
8     if flag == 1:
9         lst.append(str(i)) # i is stored in list as string
10
11 print(",".join(lst))
```

q012-01.py

Code python :

```
1 values = []
2
3 for i in range(1000, 3001):
4     s = str(i)
5     if all(int(digit) % 2 == 0 for digit in s):
6         values.append(s)
7
8 print(",".join(values))
```

q012-02.py

Code python :

```
1 def check(element):
2     return all(
3         ord(i) % 2 == 0 for i in element
4     ) # all returns True if all digits i is even in element
5
6
7 lst = [
8     str(i) for i in range(1000, 3001)
9 ] # creates list of all given numbers with string data type
10 lst = list(
11     filter(check, lst)
12 ) # filter removes element from list if check condition fails
13 print(",".join(lst))
```

q012-03.py

Code python :



```
1 lst = [str(i) for i in range(1000, 3001)]
2 lst = list(
3     filter(lambda i: all(ord(j) % 2 == 0 for j in i), lst)
4 ) # using lambda to define function inside filter function
5 print(", ".join(lst))
```

q012-04.py

### Question 13

Écrivez un programme qui accepte une phrase et qui calcule le nombre de lettres et de chiffres.

Supposons que l'entrée suivante soit fournie au programme :

Bonjour le monde !123

Ensuite, la sortie doit être :

Lettres 14

Chiffres 3

Code python :

```
1 word = input()
2 letter, digit = 0, 0
3
4 for i in word:
5     if ("a" <= i and i <= "z") or ("A" <= i and i <= "Z"):
6         letter += 1
7     if "0" <= i and i <= "9":
8         digit += 1
9
10 print("Lettres {0}\nChiffres {1}".format(letter, digit))
```

q013.py

Code python :

```
1 word = input()
2 letter, digit = 0, 0
3
4 for i in word:
5     if i.isalpha(): # returns True if alphabet
6         letter += 1
7     elif i.isnumeric(): # returns True if numeric
8         digit += 1
9 print(
10     f"Lettres {letter}\nChiffres {digit}"
11 ) # two different types of formatting method is shown in both solution
```

q013-01.py





Code python :

```
1  """ Solution by: popomaticbubble
2  """
3
4  import re
5
6  input_string = input("> ")
7  print()
8  counter = {
9      "Lettres": len(re.findall("[a-zA-Z]", input_string)),
10     "Chiffres": len(re.findall("[0-9]", input_string)),
11 }
12
13 print(counter)
```

q013-02.py

### Question 14

Écrivez un programme qui accepte une phrase et calculez le nombre de lettres en majuscules et de lettres minuscules.

Supposons que l'entrée suivante soit fournie au programme :

BonJour le Monde !

Ensuite, la sortie doit être :

Majuscules 3

Minuscules 11

Code python :

```
1  s = input()
2  d = {"Majuscules": 0, "Minuscules": 0}
3  for c in s:
4      if c.isupper():
5          d["Majuscules"] += 1
6      elif c.islower():
7          d["Minuscules"] += 1
8      else:
9          pass
10
11 print("Majuscules", d["Majuscules"])
12 print("Minuscules", d["Minuscules"])
```

q014.py

Code python :



```
1 word = input()
2 upper, lower = 0, 0
3
4 for i in word:
5     if "a" <= i and i <= "z":
6         lower += 1
7     if "A" <= i and i <= "Z":
8         upper += 1
9
10 print("Majuscules {0}\nMinuscules {1}".format(upper, lower))
```

q014-01.py

Code python :

```
1 word = input()
2 upper, lower = 0, 0
3
4 for i in word:
5     lower += i.islower()
6     upper += i.isupper()
7
8 print("UPPER CASE {0}\nLOWER CASE {1}".format(upper, lower))
```

q014-02.py

Code python :

```
1 word = input()
2 upper = sum(1 for i in word if i.isupper())
3 # sum function cumulatively sum up 1's if the condition is True
4 lower = sum(1 for i in word if i.islower())
5
6 print("UPPER CASE {0}\nLOWER CASE {1}".format(upper, lower))
```

q014-03.py

Code python :



```
1 # solution by Amitewu
2
3 string = input("Enter the sentenceBonJour le Monde! => ")
4 upper = 0
5 lower = 0
6 for x in string:
7     if x.isupper() == True:
8         upper += 1
9     if x.islower() == True:
10        lower += 1
11
12 print("UPPER CASE: ", upper)
13 print("LOWER CASE: ", lower)
```

q014-04.py

### Question 15

Écrivez un programme qui calcule la valeur d'un  $a + aa + aaa + aaaa$  avec un chiffre donné comme valeur de  $a$ .

Supposons que l'entrée suivante soit fournie au programme :

9

Ensuite, la sortie doit être :

Le résultat de :  $9 + 99 + 999 + 9999$

est : 11106

Code python :

```
1 a = input("Entrez un chiffre : ")
2 n1 = int(f"{a}")
3 n2 = int(f"{a}{a}")
4 n3 = int(f"{a}{a}{a}")
5 n4 = int(f"{a}{a}{a}{a}")
6
7 print(f"Le résultat de {a} + {a}{a} + {a}{a}{a} + {a}{a}{a}{a} est :")
8 print(f"{n1 + n2 + n3 + n4:,}".replace(",", " "))
```

q015.py

Code python :

```
1 a = input()
2 total, tmp = 0, str() # initialing an integer and empty string
3
4 for i in range(4):
5     tmp += a # concatenating 'a' to 'tmp'
6     total += int(tmp) # converting string type to integer type
7
8 print(total)
```

q015-01.py



**Code python :**

```
1 a = input()
2 total = int(a) + int(2 * a) + int(3 * a) + int(4 * a)
3 # N*a=Na, for example a="23", 2*a="2323", 3*a="232323"
4 print(total)
```

q015-02.py

**Question 16**

Utilisez une compréhension de liste pour élever au carré chaque nombre impair d'une liste. La liste est introduite par une séquence de nombres séparés par des virgules. Supposons que l'entrée suivante soit fournie au programme :

1,2,3,4,5,6,7,8,9

La sortie devrait alors être :

1,9,25,49,81

**Code python :**

```
1 values = input()
2 numbers = [str(int(x) ** 2) for x in values.split(",") if int(x) % 2 !=
↪ 0]
3 # numbers = [str(int(x) ** 2) for x in values.split(",") if int(x) % 2]
4 print(",".join(numbers))
```

q016.py

**Question 17**

Écrivez un programme qui calcule le montant net d'un compte bancaire basé sur un journal de transaction à partir de l'entrée de la console

Le format de journal des transactions est affiché comme suit :

D 100

W 200

D signifie dépôt et w retrait.

Supposons que l'entrée suivante soit fournie au programme :

D 300

D 300

W 200

D 100

Ensuite, la sortie doit être :

500

**Code python :**



```
1 netAmount = 0
2 while True:
3     s = input()
4     if not s:
5         break
6     operation, amount = s.split(" ")
7     amount = int(amount)
8     if operation == "D":
9         netAmount += amount
10    elif operation == "W":
11        netAmount -= amount
12    else:
13        pass
14
15 print(netAmount)
```

q017.py

Code python :

```
1 total = 0
2 while True:
3     s = input().split()
4     if not s: # break if the string is empty
5         break
6     cm, num = map(
7         str, s
8     ) # two inputs are distributed in cm and num in string data type
9
10    if cm == "D":
11        total += int(num)
12    if cm == "W":
13        total -= int(num)
14
15 print(total)
```

q017-01.py

Code python :



```
1 solde = 0
2 while True:
3     action = input("Dépôt/Retrait/Solde/Quitter? D/R/S/Q: ").lower()
4     if action == "d":
5         depot = input("Combien souhaitez vous déposer ? ")
6         solde = solde + int(depot)
7     elif action == "r":
8         retrait = input("Combien souhaitez vous retirer ? ")
9         solde = solde - int(retrait)
10    elif action == "s":
11        print(solde)
12    else:
13        quit()
```

q017-02.py

### Question 18

Un site Web oblige les utilisateurs à saisir le nom d'utilisateur et le mot de passe pour s'inscrire.Écrivez un programme pour vérifier la validité de la saisie du mot de passe par les utilisateurs.

Voici les critères de vérification du mot de passe :

- ① Au moins 1 lettre entre [a-z]
- ② Au moins 1 nombre entre [0-9]
- ③ Au moins 1 lettre entre [A-Z]
- ④ Au moins 1 personnage de [\$ # @]
- ⑤ Longueur minimal : 6
- ⑥ Longueur maximale : 12
- ⑦ Ne doit pas contenir d'espace

Votre programme doit accepter une séquence de mots de passe séparés par des virgules et les vérifiera conformément aux critères ci-dessus.Les mots de passe qui correspondent aux critères doivent être imprimés, chacun séparé par une virgule.

Exemple

Si les mots de passe suivants sont donnés en entrée au programme :

ABd1234@1,a F1#,2w3E\*,2We3345

Ensuite, la sortie du programme doit être :

AbD1234@1

**Code python :**



```
1 import re
2
3 def check_password_validity(password):
4     if (6 <= len(password) <= 12 and
5         re.search("[a-z]", password) and
6         re.search("[0-9]", password) and
7         re.search("[A-Z]", password) and
8         re.search("[$#@]", password) and
9         not re.search("\s", password)):
10         return True
11     return False
12
13 input_passwords = input("Entrez une séquence de mots de passe séparés
    ↪ par des virgules : ")
14 passwords = input_passwords.split(',')
15
16 valid_passwords = [password for password in passwords if
    ↪ check_password_validity(password)]
17
18 print(",".join(valid_passwords))
```

q018.py

**Code python :**

```
1 import re
2 value = []
3 items=[x for x in input().split(',')]
4 for p in items:
5     if len(p)<6 or len(p)>12:
6         continue
7     else:
8         pass
9     if not re.search("[a-z]",p):
10         continue
11     elif not re.search("[0-9]",p):
12         continue
13     elif not re.search("[A-Z]",p):
14         continue
15     elif not re.search("[$#@]",p):
16         continue
17     elif re.search("\s",p):
18         continue
19     else:
20         pass
21     value.append(p)
22 print(",".join(value))
```

q018-01.py



Code python :

```
1 def is_low(x): # Returns True if the string has a lowercase
2     for i in x:
3         if "a" <= i and i <= "z":
4             return True
5     return False
6
7
8 def is_up(x): # Returns True if the string has a uppercase
9     for i in x:
10        if "A" <= i and i <= "Z":
11            return True
12    return False
13
14
15 def is_num(x): # Returns True if the string has a numeric digit
16     for i in x:
17         if "0" <= i and i <= "9":
18             return True
19    return False
20
21
22 def is_other(x): # Returns True if the string has any "$#@
23     for i in x:
24         if i == "$" or i == "#" or i == "@":
25             return True
26    return False
27
28
29 s = input().split(",")
30 lst = []
31
32 for i in s:
33     length = len(i)
34     if (
35         6 <= length
36         and length <= 12
37         and is_low(i)
38         and is_up(i)
39         and is_num(i)
40         and is_other(i)
41     ): # Checks if all the requirments are fulfilled
42         lst.append(i)
43
44 print(",".join(lst))
```

q018-02.py





Code python :

```
1 def check(x):
2     cnt = 6 <= len(x) and len(x) <= 12
3     for i in x:
4         if i.isupper():
5             cnt += 1
6             break
7     for i in x:
8         if i.islower():
9             cnt += 1
10            break
11    for i in x:
12        if i.isnumeric():
13            cnt += 1
14            break
15    for i in x:
16        if i == "@" or i == "#" or i == "$":
17            cnt += 1
18            break
19    return cnt == 5
20
21
22 # counting if total 5 all conditions are fulfilled then returns True
23
24
25 s = input().split(",")
26 lst = filter(check, s)
27 # Filter function pick the words from s, those returns True by check()
28 ↪ function
29 print(",".join(lst))
```

q018-03.py

Code python :



```
1 import re
2
3 s = input().split(',')
4 lst = []
5
6 for i in s:
7     cnt = 0
8     cnt+=(6<=len(i) and len(i)<=12)
9     cnt+=bool(re.search("[a-z]",i))      # here re module includes a
    ↪ function re.search() which returns the object information
10    cnt+=bool(re.search("[A-Z]",i))      # of where the pattern string
    ↪ i is matched with any of the [a-z]/[A-z]/[0=9]/[@#$] characters
11    cnt+=bool(re.search("[0-9]",i))      # if not a single match found
    ↪ then returns NONE which converts to False in boolean
12    cnt+=bool(re.search("[@#$]",i))      # expression otherwise True if
    ↪ found any.
13    if cnt == 5:
14        lst.append(i)
15
16 print(",".join(lst))
```

q018-04.py

### Code python :

```
1 import re
2 a = input('Enter passwords: ').split(',')
3 pass_pattern =
    ↪ re.compile(r"^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%&])\S{6,12}$")
4 for i in a:
5     if pass_pattern.fullmatch(i):
6         print(i)
```

q018-05.py

### Question 19

Vous devez rédiger un programme pour trier les tuples (nom, âge, hauteur) par ordre croissant où le nom est une chaîne, l'âge et la taille sont des entiers. Les tuples sont entrés par console.

Les critères de tri sont :

- ① Trier basé sur le nom ;
- ② puis trier en fonction de l'âge ;
- ③ Puis triez par la taille.

Si les tuples suivants sont donnés comme entrée au programme :

Tom,19,80



John,20,90

Jony,17,91

Jony,17,93

Json,21,85 Ensuite, la sortie du programme doit être :

[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]

Code python :

```
1 from operator import itemgetter
2
3 result = []
4 while True:
5     s = input()
6     if not s:
7         break
8     result.append(tuple(s.split(",")))
9
10 print(sorted(result, key=itemgetter(0, 1, 2)))
```

q019.py

Code python :

```
1 def sort_tuples(tuples_list):
2     # Trier les tuples par nom, puis par âge, puis par taille
3     return sorted(tuples_list, key=lambda x:(x[0], int(x[1]),
4         ↪ int(x[2])))
5
6 # Entrée des tuples par la console
7 input_data = """Tom,19,80
8 John,20,90
9 Jony,17,91
10 Jony,17,93
11 Json,21,85"""
12
13 # Conversion des données d'entrée en une liste de tuples
14 tuples_list = [tuple(item.split(',')) for item in
15     ↪ input_data.split('\n')]
16
17 # Tri des tuples
18 sorted_tuples = sort_tuples(tuples_list)
19
20 # Affichage du résultat
21 print(sorted_tuples)
```

q019-01.py

Question 20





Définissez une classe avec un générateur qui peut itérer les nombres, qui sont divisibles par 7, entre une plage donnée 0 et n. Par exemple l'entrée suivante :

18

donne la sortie :

0

7

14

Code python :

```
1 class DivisibleBySeven:
2     def __init__(self, n):
3         self.n = n
4
5     def generator(self):
6         for i in range(0, self.n + 1):
7             if i % 7 == 0:
8                 yield i
9
10 # Exemple d'utilisation
11 n = int(input("Entrez la valeur de n : "))
12 divisible_by_seven = DivisibleBySeven(n)
13
14 for number in divisible_by_seven.generator():
15     print(number)
16
```

q020.py

Code python :

```
1 class MyGen():
2     def by_seven(self, n):
3         for i in range(0, int(n/7) + 1):
4             yield i * 7
5
6 for i in MyGen().by_seven( int(input('Please enter a number... ')) ):
7     print(i)
```

q020-01.py

Code python :



```
1 class Divisible:
2
3     def by_seven(self, n):
4         for number in range(1, n + 1):
5             if number % 7 == 0:
6                 yield number
7
8
9 divisible = Divisible()
10 generator = divisible.by_seven(int(input("Please insert a number. -->
    ↵ ")))
11 for number in generator:
12     print(number)
```

q020-02.py

### Question 21

Un robot se déplace dans un avion à partir du point d'origine (0,0). Le robot peut se déplacer vers le haut, le bas, la gauche et la droite.

La trace du mouvement du robot est indiquée comme suit :

UP 5

DOWN 3

LEFT 3

RIGHT 2

Les nombres qui suivent la direction sont des pas.

Veillez écrire un programme pour calculer la distance entre la position actuelle après une séquence de mouvements et le point d'origine. Si la distance est un flottant, il suffit d'imprimer l'entier le plus proche.

Exemple :

Si les tuples suivants sont donnés comme entrée au programme : UP 5

DOWN 3

LEFT 3

RIGHT 2

Ensuite, la sortie du programme doit être :

2

**Code python :**



```
1 import math
2
3 x, y = 0, 0
4 while True:
5     s = input().split()
6     if not s:
7         break
8     if s[0] == "UP": # s[0] indicates command
9         x -= int(s[1]) # s[1] indicates unit of move
10    if s[0] == "DOWN":
11        x += int(s[1])
12    if s[0] == "LEFT":
13        y -= int(s[1])
14    if s[0] == "RIGHT":
15        y += int(s[1])
16 # N**P means N^P
17
18 dist = round(math.sqrt(x**2 + y**2))
19 # euclidean distance = square root of (x^2+y^2) and rounding it to
    ↪ nearest integer
20 print(dist)
```

q021.py

### Code python :

```
1 '''Solution by: pratikb0501
2 '''
3
4 from math import sqrt
5 lst = []
6 position = [0,0]
7 while True:
8     a = input()
9     if not a:
10         break
11     lst.append(a)
12 for i in lst:
13     if 'UP' in i:
14         position[0] -= int(i.strip('UP '))
15     if 'DOWN' in i:
16         position[0] += int(i.strip('DOWN '))
17     if 'LEFT' in i:
18         position[1] -= int(i.strip('LEFT '))
19     if 'RIGHT' in i:
20         position[1] += int(i.strip('RIGHT '))
21 print(round(sqrt(position[1]**2 + position[0]**2)))
```

q021-01.py

**Question 22**

Écrivez un programme pour calculer la fréquence des mots à partir de l'entrée. La sortie doit sortir après le tri de la clé de manière alphanumérique.

Supposons que l'entrée suivante soit fournie au programme :

Nouveau sur Python ou choisir entre Python 2 et Python 3 ? Lisez Python 2 ou Python 3.

Ensuite, la sortie doit être :

```
2 :2
3 :1
3. :1
? :1
Lisez :1
Nouveau :1
Python :5
choisir :1
entre :1
et :1
ou :2
sur :1
```

Code python :

```
1 ss = input().split()
2 word = sorted(set(ss))
3 # split words are stored and sorted as a set
4
5 for i in word:
6     print("{0}:{1}".format(i, ss.count(i)))
```

q022.py

Code python :

```
1 ss = input().split()
2 dict = {}
3 for i in ss:
4     i = dict.setdefault(i, ss.count(i))
5     # setdefault() function takes key & value to set it as dictionary.
6
7 dict = sorted(dict.items())
8 # items() function returns both key & value of dictionary as a list
9 # and then sorted. The sort by default occurs in order of 1st -> 2nd
10 ↪ key
11 for i in dict:
12     print(f"{i[0]}:{i[1]}")
```

q022-01.py



Code python :

```
1 ss = input().split()
2 dict = {i: ss.count(i) for i in ss}
3 # sets dictionary as i-> split word & ss.count(i) -> total occurrence
  ↪ of i in ss
4 dict = sorted(dict.items())
5 # items() function returns both key & value of dictionary as a list
6 # and then sorted. The sort by default occurs in order of 1st -> 2nd
  ↪ key
7 for i in dict:
8     print(f"{i[0]}:{i[1]}")
```

q022-02.py

Code python :

```
1 from collections import Counter
2
3 ss = input().split()
4 ss = Counter(ss)
5 # returns key & frequency as a dictionary
6 ss = sorted(ss.items())
7 # returns as a tuple list
8
9 for i in ss:
10     print("%s:%d" % (i[0], i[1]))
```

q022-03.py

Code python :

```
1 from pprint import pprint
2
3 p = input().split()
4 pprint({i: p.count(i) for i in p})
```

q022-04.py

### Question 23

Écrire une fonction qui peut calculer la valeur carrée d'un nombre.

Code python :

```
1 def square(num):
2     return num ** 2
3
4 print(square(2))
5 print(square(3))
```

q023.py



**Question 24**

Python possède de nombreuses fonctions intégrées, il a une fonction de documentation intégrée pour toutes ses fonctions. Veuillez écrire un programme pour imprimer la documentation des fonctions suivantes :

- `abs()`
- `int()`
- `input ()`

Puis écrire une fonction qui peut calculer la valeur carrée d'un nombre et lui ajouter une documentation.

**Code python :**

```
1 print(abs.__doc__)
2 print(int.__doc__)
3 print(input.__doc__)
4
5 def square(num):
6     '''Return the square value of the input number.
7
8     The input number must be integer.
9     '''
10    return num ** 2
11
12
13 print(square(2))
14 print(square.__doc__)
```

q024.py

**Question 25**

Définir une classe qui a un paramètre de classe et un même paramètre d'instance.

**Indices :**

- Pour définir un paramètre d'instance, il faut l'ajouter dans la méthode `__init__`.
- Vous pouvez initialiser un objet avec un paramètre de construction ou en définir la valeur ultérieurement.

**Code python :**



```
1 class Person:
2     # Define the class parameter "name"
3     name = "Nom non attribué"
4
5     def __init__(self, name = None):
6         if name is None:
7             self.name = self.name
8         else:
9             self.name = name
10
11 jeffrey = Person("Jeffrey")
12 print(f"Person.name : {Person.name} et jeffrey.name : {jeffrey.name}")
13
14
15 nico = Person()
16 print(f"Person.name : {Person.name}, nico.name : {nico.name}")
17 nico.name = "Nico"
18 print(f"Person.name : {Person.name}, nico.name : {nico.name}")
```

q025.py

Code python :

```
1 class Person:
2     # Define the class parameter "name"
3     name = "Person"
4
5     def __init__(self, name = None):
6         # self.name is the instance parameter
7         self.name = name
8
9 jeffrey = Person("Jeffrey")
10 print ("%s name is %s" % (Person.name, jeffrey.name))
11
12 nico = Person()
13 nico.name = "Nico"
14 print ("%s name is %s" % (Person.name, nico.name))
```

q025-01.py

Code python :



```
1 class Car:
2     name = "Car"
3
4     def __init__(self, name = None):
5         self.name = name
6
7 honda=Car("Honda")
8 print("%s name is %s"%(Car.name, honda.name))
9
10 toyota=Car()
11 toyota.name="Toyota"
12 print("%s name is %s"%(Car.name, toyota.name))
```

q025-02.py

### Question 26

Définissez une fonction qui peut calculer la somme de deux nombres.

**Indices :** Définissez une fonction avec deux nombres comme arguments. Vous pouvez calculer la somme dans la fonction et renvoyer la valeur.

Code python :

```
1 def SumFunction(number1, number2):
2     return number1+number2
3
4 print(SumFunction(1, 2))
```

q026.py

### Question 27

Définissez une fonction qui peut convertir un entier en une chaîne et l'imprimer dans la console.

**Indices :** Utilisez STR () pour convertir un nombre en chaîne.

Code python :

```
1 def printValue(n):
2     print(str(n))
3
4 printValue(3)
```

q027.py

### Question 28

Définir une fonction qui peut recevoir deux nombres entiers sous forme de chaîne de caractères et calculer leur somme, puis l'imprimer dans la console.



**Indices :** Utilisez `int()` pour convertir une chaîne en entier.

**Code python :**

```
1 def printValue(s1,s2):  
2     print(int(s1)+int(s2))  
3  
4 printValue("3","4") #7
```

q028.py

### Question 29

Définissez une fonction qui peut accepter deux chaînes en entrée et les concaténer, puis l'imprimer dans la console.

**Indices :** Utiliser `+` pour concaténer les chaînes

**Code python :**

```
1 def printValue(s1,s2):  
2     print(s1+s2)  
3  
4 printValue("3","4") #34
```

q029.py

### Question 30

Définir une fonction capable d'accepter deux chaînes de caractères en entrée et d'imprimer la chaîne de caractères de longueur maximale dans la console. Si les deux chaînes ont la même longueur, la fonction doit imprimer les deux une par ligne.

**Indices :** Utilisez la fonction `Len()` pour obtenir la longueur d'une chaîne

**Code python :**

```
1 def printValue(s1, s2):  
2     len1 = len(s1)  
3     len2 = len(s2)  
4     if len1 > len2:  
5         print(s1)  
6     elif len2 > len1:  
7         print(s2)  
8     else:  
9         print(s1)  
10        print(s2)  
11  
12  
13 printValue("one","three")  
14 print()  
15 printValue("five","four")
```

q030.py

**Question 31**

Définir une fonction qui accepte un nombre entier en entrée et qui imprime "C'est un nombre pair" si le nombre est pair, sinon "C'est un nombre impair".

**Indices :** Utilisez un opérateur % pour vérifier si un nombre est pair ou impair.

**Code python :**

```
1 def checkValue(n):
2     if n % 2 == 0:
3         print("C'est un nombre pair")
4     else:
5         print("C'est un nombre impair")
6
7
8 checkValue(7)
9 checkValue(8)
```

q031.py

**Question 32**

Définir une fonction capable d'imprimer un dictionnaire dont les clés sont des nombres compris entre 1 et 3 (les deux inclus) et dont les valeurs sont des carrés des clés.

**Indices :**

- Utiliser le modèle dict[key]=value pour placer une entrée dans un dictionnaire.
- Utiliser l'opérateur \*\* pour obtenir la puissance d'un nombre.

**Code python :**

```
1 def printDict():
2     d = {}
3     for i in range(1, 4):
4         d[i] = i**2
5     print(d)
6
7
8 printDict()
```

q032.py

**Code python :**



```
1 def printDict():
2     d = {}
3     d[1] = 1
4     d[2] = 2**2
5     d[3] = 3**2
6     print(d)
7
8
9 printDict()
```

q032-01.py

Code python :

```
1 def printDict():
2     print({x: x**2 for x in range(1, 4)})
3
4
5 printDict()
```

q032-02.py

### Question 33

Définir une fonction capable d'imprimer un dictionnaire dont les clés sont des nombres compris entre 1 et 20 (les deux inclus) et dont les valeurs sont des carrés de clés.

Indices :

- Utiliser le modèle `dict[key]=value` pour placer une entrée dans un dictionnaire.
- Utiliser l'opérateur `**` pour obtenir la puissance d'un nombre.
- Utiliser `range()` pour les boucles.

Code python :

```
1 def printDict():
2     d = dict()
3     for i in range(1, 21):
4         d[i] = i**2
5     print(d)
6
7
8
9 printDict()
```

q033.py

Code python :



```
1 def printDict():
2     print({x: x**2 for x in range(1, 21)})
3
4
5
6 printDict()
```

q033-01.py

### Question 34

Définir une fonction capable de générer un dictionnaire dont les clés sont des nombres compris entre 1 et 20 (les deux inclus) et dont les valeurs sont des carrés de clés. La fonction ne doit imprimer que les valeurs.

**Indices :**

- Utiliser le modèle `dict[key]=value` pour placer une entrée dans un dictionnaire.
- Utiliser l'opérateur `**` pour obtenir la puissance d'un nombre.
- Utiliser `range()` pour les boucles.
- Utiliser `values()` pour itérer les clés dans le dictionnaire. Nous pouvons également utiliser `items()` pour obtenir des paires clé/valeur.

**Code python :**

```
1 def printDict():
2     d = dict()
3     for i in range(1, 21):
4         d[i] = i**2
5     for v in d.values():
6         print(v)
7
8
9 printDict()
```

q034.py

### Question 35

Définir une fonction capable de générer un dictionnaire dont les clés sont des nombres compris entre 1 et 20 (les deux inclus) et dont les valeurs sont des carrés de clés. La fonction ne doit imprimer que les clés.

**Indices :**

- Utiliser le modèle `dict[key]=value` pour placer une entrée dans un dictionnaire.
- Utiliser l'opérateur `**` pour obtenir la puissance d'un nombre.
- Utiliser `range()` pour les boucles.



- Utiliser `keys()` pour itérer les clés dans le dictionnaire. Nous pouvons également utiliser `items()` pour obtenir des paires clé/valeur.

Code python :

```
1 def printDict():
2     d = dict()
3     for i in range(1, 21):
4         d[i] = i**2
5     for k in d.keys():
6         print(k)
7
8
9 printDict()
```

q035.py

### Question 36

Définir une fonction capable de générer et d'imprimer une liste dont les valeurs sont des carrés de nombres compris entre 1 et 20 (les deux inclus).

Indices :

- Utilisez `**` Opérateur pour obtenir la puissance d'un nombre.
- Utilisez la `range()` pour les boucles.
- Utilisez `list.append()` pour ajouter des valeurs dans une liste.

Code python :

```
1 def printList():
2     li = list()
3     for i in range(1, 21):
4         li.append(i**2)
5     print(li)
6
7
8 printList()
```

q036.py

### Question 37

Définir une fonction capable de générer une liste dont les valeurs sont des carrés de nombres compris entre 1 et 20 (les deux inclus). La fonction doit ensuite imprimer les 5 derniers éléments de la liste.

Indices :

- Utilisez `**` Opérateur pour obtenir la puissance d'un nombre.
- Utilisez la `range()` pour les boucles.
- Utilisez `list.append()` pour ajouter des valeurs dans une liste.
- Utilisez `[N1 : N2]` pour slicer une liste





Code python :

```
1 def printList():
2     li = []
3     for i in range(1, 21):
4         li.append(i**2)
5     print(li[-5:])
6
7
8 printList()
```

q037.py

### Question 38

Définir une fonction capable de générer et d'imprimer un tuple dont les valeurs sont des carrés de nombres compris entre 1 et 20 (les deux inclus).

Indices :

- Utilisez \*\* Opérateur pour obtenir la puissance d'un nombre.
- Utilisez la range() pour les boucles.
- Utilisez list.append() pour ajouter des valeurs dans une liste.
- Utilisez tuple() pour obtenir un tuple d'une liste.

Code python :

```
1 def printTuple():
2     li = []
3     for i in range(1, 21):
4         li.append(i**2)
5     print(tuple(li))
6
7
8 printTuple()
```

q038.py

Code python :

```
1 def printTuple():
2     t = tuple(i**2 for i in range(1, 21))
3     print(t)
4
5
6 printTuple()
```

q038-01.py

### Question 39

Ecrivez un programme pour générer et imprimer un autre tuple dont les valeurs sont des nombres pairs dans le tuple donné (1,2,3,4,5,6,7,8,9,10).



**Indices :**

- Utilisez "for" pour itérer le tuple
- Utilisez Tuple() pour générer un tuple à partir d'une liste.

**Code python :**

```
1 tp = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2 li = []
3 for i in tp:
4     if i % 2 == 0:
5         li.append(i)
6
7 tp2 = tuple(li)
8 print(tp2)
```

q039.py

**Code python :**

```
1 tpl = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2 tpl1 = tuple(i for i in tpl if i % 2 == 0)
3 print(tpl1)
```

q039-01.py

**Code python :**

```
1 tpl = (1,2,3,4,5,6,7,8,9,10)
2 tpl1 = tuple(filter(lambda x : x%2==0,tpl)) # Lambda function returns
↳ True if found even element.
3
# Filter removes data for
↳ which function returns
↳ False
4 print(tpl1)
```

q039-02.py

**Question 40**

Écrire un programme qui accepte une chaîne de caractères en entrée pour imprimer "Oui" si la chaîne est "oui" ou "OUI" ou "Oui", sinon imprimer "Non".

**Code python :**

```
1 s = input()
2 if s.upper() == "YES":
3     print("Yes")
4 else:
5     print("No")
```

q040.py



### Question 41

Écrivez un programme qui peut filtrer les nombres pairs dans une liste en utilisant la fonction filter.

La liste est la suivante : [1,2,3,4,5,6,7,8,9,10].

**Indices :**

- Utilisez filter() pour filtrer certains éléments dans une liste.
- Utilisez lambda pour définir des fonctions anonymes.

**Code python :**

```
1 li = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 evenNumbers = filter(lambda x: x % 2 == 0, li)
3
4 print(list(evenNumbers))
```

q041.py

### Question 42

Écrivez un programme qui peut utiliser map() pour créer une liste dont les éléments sont le carré des éléments de [1,2,3,4,5,6,7,8,9,10].

**Indices :**

- Utilisez map() pour générer une liste.
- Utilisez lambda pour définir des fonctions anonymes.

**Code python :**

```
1 li = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 squaredNumbers = map(lambda x: x**2, li)
3 print(list(squaredNumbers))
```

q042.py

### Question 43

Écrivez un programme qui peut utiliser map() et filter() pour créer une liste dont les éléments sont les carrés des nombres pairs de la liste :

[1,2,3,4,5,6,7,8,9,10].

**Indices :**

- Utilisez map() pour générer une liste.
- Utilisez filter() pour filtrer les éléments d'une liste.
- Utilisez lambda pour définir des fonctions anonymes.

**Code python :**



```
1 li = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 evenNumbers = map(lambda x: x**2, filter(lambda x: x % 2 == 0, li))
3 print(list(evenNumbers))
```

q043.py

#### Question 44

Écrivez un programme qui peut filtrer() pour faire une liste dont les éléments sont des nombres pairs entre 1 et 20 (les deux inclus).

**Indices :** Utilisez filter() pour filtrer les éléments d'une liste. Utilisez lambda pour définir des fonctions anonymes.

Code python :

```
1 evenNumbers = filter(lambda x: x % 2 == 0, range(1, 21))
2 print(list(evenNumbers))
```

q044.py

Code python :

```
1 def even(x):
2     return x%2==0
3
4 evenNumbers = filter(even, range(1,21))
5 print(list(evenNumbers))
```

q044-01.py

#### Question 45

Écrivez un programme qui peut utiliser map() pour créer une liste dont les éléments sont des carrés de nombres compris entre 1 et 20 (les deux inclus).

**Indices :** Utilisez map() pour générer une liste. Utilisez lambda pour définir des fonctions anonymes.

Code python :

```
1 squaredNumbers = map(lambda x: x**2, range(1, 21))
2 print(list(squaredNumbers))
```

q045.py

#### Question 46

Définissez une classe nommée American qui possède une méthode statique appelée printNationality.

**Indices :** Utilisez @StaticMethod Decorator pour définir la méthode statique de classe.



Code python :

```
1 class American():
2     @staticmethod
3     def printNationality():
4         print("I am American")
5
6 american = American()
7 american.printNationality()    # this will not run if @staticmethod does
    ↪ not decorates the function.
8                               # Because the class has no instance.
9
10 American.printNationality()   # this will run even though the
    ↪ @staticmethod
11                               # does not decorate printNationality()
```

q046.py

### Question 47

Définissez une classe nommée American et sa sous-classe Newyorker.

Code python :

```
1 class American():
2     pass
3
4 class NewYorker(American):
5     pass
6
7 american = American()
8 newyorker = NewYorker()
9
10 print(american)
11 print(newyorker)
```

q047.py

Code python :



```
1 class American:
2     def __init__(self, name):
3         self.name = name
4
5     def describe(self):
6         return f"{self.name} is an American."
7
8
9 class NewYorker(American):
10    def describe(self):
11        parent_description = super().describe()
12        return f"{parent_description} Specifically, {self.name} is a
    ↪ New Yorker."
13
14
15 # Exemple d'utilisation
16 anAmerican = American("John")
17 aNewYorker = NewYorker("Jane")
18
19 print(anAmerican.describe()) # Affichera "John is an American."
20 print(
21     aNewYorker.describe()
22 ) # Affichera "Jane is an American. Specifically, Jane is a New
    ↪ Yorker."
```

q047-01.py

### Question 48

Définir une classe nommée Cercle qui peut être construite par un rayon. La classe Cercle possède une méthode qui permet de calculer la surface.

Puis définir une classe rectangle qui peut être construit par une longueur et une largeur. La classe Rectangle possède une méthode qui permet de calculer la surface.

**Indices :** Utilisez `Def nom_de_le_methode(Self)` pour définir une méthode.

**Code python :**

```
1 class Circle(object):
2     def __init__(self, r):
3         self.radius = r
4
5     def area(self):
6         return self.radius**2 * 3.14
7
8
9 aCircle = Circle(2)
10 print(aCircle.area())
```

q048.py



Code python :

```
1 class Rectangle():
2     def __init__(self,l,w):
3         self.length = l
4         self.width = w
5
6     def area(self):
7         return self.length*self.width
8
9
10 rect = Rectangle(2,4)
11 print(rect.area())
```

q048-01.py

### Question 49

Définissez une classe nommée Shape et sa sous-classe Square. La classe Square possède une fonction init qui prend une longueur en argument. Les deux classes disposent d'une fonction area qui permet d'imprimer l'aire de la forme, l'aire de Shape étant égale à 0 par défaut.

**Indices :** Pour remplacer une méthode dans une super-classe, nous pouvons définir une méthode portant le même nom dans la super-classe.

Code python :

```
1 class Shape():
2     def __init__(self):
3         pass
4
5     def area(self):
6         return 0
7
8 class Square(Shape):
9     def __init__(self,length = 0):
10         Shape.__init__(self)
11         self.length = length
12
13     def area(self):
14         return self.length*self.length
15
16 Asqr = Square(5)
17 print(Asqr.area())      # prints 25 as given argument
18
19 print(Square().area())  # prints zero as default area
```

q048bis.py

**Question 50**

En supposant que nous avons des adresses e-mail au format **username@companyname.com**, veuillez écrire un programme pour imprimer le nom d'utilisateur d'une adresse e-mail donnée. Les noms d'utilisateurs et les noms d'entreprise sont composés de lettres uniquement.

Exemple :

Si l'adresse e-mail suivante est donnée comme entrée au programme :

John@google.com

Ensuite, la sortie du programme doit être :

John

**Indices :** aidez vous du package "re"

**Code python :**

```
1 email = "john@google.com"
2 email = email.split('@')
3 print(email[0])
```

q049.py

**Code python :**

```
1 import re
2
3 email = "john@google.com elise@python.com"
4 pattern = "(\w+)\w+.com"
5 ans = re.findall(pattern, email)
6 print(ans)
```

q049-01.py

**Question 51**

Écrivez un programme qui accepte une séquence de mots séparés par des espaces comme entrée et qui génère une liste contenant toutes les valeurs numériques de cette entrée.

Exemple :

Si les mots suivants sont donnés en entrée au programme :

2 chats et 3 chiens.

Ensuite, la sortie du programme doit être :





`['2', '3']`

**Indices :** Utilisez `re.findall()` pour trouver tous les sous-chaînes à l'aide de regex.

**Code python :**

```
1 phrase = input().split()
2 ans = []
3 for word in phrase:
4     if word.isdigit():      # can also use isnumeric() / isdecimal()
        ↪ function instead
5     ans.append(word)
6 print(ans)
```

q050.py

**Code python :**

```
1 phrase = input().split()
2 ans = [word for word in phrase if word.isdigit()] # using list
        ↪ comprehension method
3 print(ans)
```

q050-01.py

**Code python :**

```
1 import re
2
3 phrase = input()
4 pattern = "\d+"
5 ans = re.findall(pattern, phrase)
6 print(ans)
```

q050-02.py

### Question 52

Écrivez un programme pour calculer :

$f(n) = f(n - 1) + 100$  quand  $n > 0$

et  $f(0) = 1$

avec une entrée  $n$  donnée par console ( $n > 0$ ).

Exemple :

Si le  $n$  suivant est donné en entrée au programme :

5

Ensuite, la sortie du programme doit être :

500



**Indices :** Nous pouvons définir une fonction récursive dans Python.

**Code python :**

```
1 def f(n):  
2     if n == 0:  
3         return 0  
4     return f(n - 1) + 100  
5  
6  
7 n = int(input())  
8 print(f(n))
```

q052.py

**Code python :**

```
1 n = int(input())  
2 f = lambda x: f(x - 1) + 100 if x > 0 else 0  
3 print(f(n))
```

q052-01.py

### Question 53

La séquence Fibonacci est calculée en fonction de la formule suivante :

$$f(n) = 0 \text{ si } n = 0$$

$$f(n) = 1 \text{ si } n = 1$$

$$f(n) = f(n - 1) + f(n - 2) \text{ si } n > 1$$

Veuillez écrire un programme pour calculer la valeur de  $F(n)$  avec une entrée  $n$  donnée par console.

Exemple :

Si le  $n$  suivant est donné en entrée au programme :

7

Ensuite, la sortie du programme doit être :

13

**Indices :** Nous pouvons définir une fonction récursive dans Python.

**Code python :**



```
1 def f(n):
2     if n == 0:
3         return 0
4     elif n == 1:
5         return 1
6     else:
7         return f(n - 1) + f(n - 2)
8
9
10 n = int(input())
11 print(f(n))
```

q053.py

Code python :

```
1 def f(n):
2     if n < 2:
3         return n
4     return f(n - 1) + f(n - 2)
5
6
7 n = int(input())
8 print(f(n))
```

q053-01.py

Code python :

```
1 def f(n):
2     return n if n <= 1 else f(n - 1) + f(n - 2)
3
4
5 n = int(input("Entrez un nombre : "))
6 print(f(n))
```

q053-02.py

Code python :

```
1 n = int(input())
2 f = lambda x: 0 if x == 0 else 1 if x == 1 else f(x - 1) + f(x - 2)
3 print(",".join([str(f(x)) for x in range(0, n + 1)]))
```

q053-03.py

### Question 54

La séquence Fibonacci est calculée en fonction de la formule suivante :

$$f(n) = 0 \text{ si } n = 0$$
$$f(n) = 1 \text{ si } n = 1$$



$$f(n) = f(n - 1) + f(n - 2) \text{ si } n > 1$$

Veillez écrire un programme en utilisant la compréhension de la liste pour imprimer la séquence Fibonacci sous forme de virgule séparée avec une entrée N donnée par console.

Exemple :

Si le n suivant est donné en entrée au programme :

7

Ensuite, la sortie du programme doit être :

0,1,1,2,3,5,8,13

**Indices :**

- Nous pouvons définir une fonction récursive dans Python.
- Utilisez la compréhension de la liste pour générer une liste à partir d'une liste existante.
- Utilisez <string>.Join() pour concaténer une liste de chaînes.

**Code python :**

```
1 def f(n):
2     if n == 0:
3         return 0
4     if n == 1:
5         return 1
6     return f(n - 1) + f(n - 2)
7
8
9 n = int(input())
10 values = [str(f(x)) for x in range(0, n + 1)]
11 print(",".join(values))
```

q054.py

**Code python :**



```
1 def f(n):
2     if n < 2:
3         fibo[n] = n
4         return fibo[n]
5     fibo[n] = f(n - 1) + f(n - 2)
6     return fibo[n]
7
8
9 n = int(input())
10 fibo = [0] * (n + 1)
11 # initialize a list of size (n+1)
12 f(n)
13 # call once and it will set value to fibo[0-n]
14 fibo = [str(i) for i in fibo]
15 # converting integer data to string type
16 ans = ",".join(fibo)
17 # joining all string element of fibo with ',' character
18 print(ans)
```

q054-01.py

Code python :

```
1 def fibo(n):
2     if n < 2:
3         return n
4     return fibo(n - 1) + fibo(n - 2)
5
6
7 def print_fiblist(n):
8     fib_list = [(str(fibo(i))) for i in range(0, n + 1)]
9     return print(",".join(fib_list))
10
11
12 n = int(input())
13 print_fiblist(n)
```

q054-02.py

Code python :



```
1 def question_62(n):
2     if n == 0:
3         return [0]
4     if n == 1:
5         return [0, 1]
6     sequence = [0, 1]
7     a, b = 0, 1
8     for x in range(2, n + 1):
9         c = a + b
10        sequence.append(c)
11        a = b
12        b = c
13    return sequence
14
15
16 print(question_62(10))
```

q054-03.py

### Question 55

Écrire un programme à l'aide du générateur pour imprimer les nombres pair entre 0 et N sous forme d'une suite de valeur séparées par des virgules. La valeur N est fournie par l'utilisateur.

Exemple :

Si la valeur de N est :

10

La sortie du programme doit être :

0,2,4,6,8,10

**Indices :** Utilisez yield pour produire la valeur suivante dans le générateur.

**Code python :**



```
1 def EvenGenerator(n):
2     i = 0
3     while i <= n:
4         if i % 2 == 0:
5             yield i
6             i += 1
7
8
9 n = int(input())
10 values = []
11 for i in EvenGenerator(n):
12     values.append(str(i))
13
14 print(",".join(values))
```

q055.py

Code python :

```
1 # Solution by: StartZero
2 n = int(input())
3
4 for i in range(0, n + 1, 2):
5     if i < n - 1:
6         print(i, end=",")
7     else:
8         print(i)
```

q055-01.py

### Question 56

Veillez écrire un programme utilisant un générateur pour imprimer les nombres divisibles par 5 et 7 entre 0 et n sous la forme d'une liste séparée par des virgules. La valeur n est fournie par l'utilisateur.

Exemple :

Si le n suivant est donné en entrée au programme :

100

Ensuite, la sortie du programme doit être :

0,35,70

**Indices :** Utilisez le yield pour produire la valeur suivante dans le générateur.

Code python :



```
1 def NumGenerator(n):
2     for i in range(n + 1):
3         if i % 5 == 0 and i % 7 == 0:
4             yield i
5
6
7 n = int(input())
8 values = []
9 for i in NumGenerator(n):
10     values.append(str(i))
11
12 print(",".join(values))
```

q056.py

Code python :

```
1 def generate(n):
2     for i in range(n + 1):
3         if (
4             i % 35 == 0
5         ): # 5*7 = 35, if a number is divisible by a & b then it is
6             ↪ also divisible by a*b
7             yield i
8
9 n = int(input())
10 resp = [str(i) for i in generate(n)]
11 print(",".join(resp))
```

q056-01.py

### Question 57

Écrire un code pour vérifier que tous les nombres de la liste [2,4,6,8] sont pairs.

**Indices :** Utilisez "assert expression" pour effectuer l'opération.

Code python :

```
1 li = [2, 4, 6, 8]
2 for i in li:
3     assert i % 2 == 0
```

q057.py

### Question 58

Veuillez écrire une fonction de recherche binaire qui recherche un élément dans une liste triée. La fonction doit renvoyer l'index de l'élément à rechercher dans la liste.

**Indices :** Utilisez if / elif pour gérer les conditions.





Code python :

```
1 import math
2
3
4 def bin_search(li, element):
5     bottom = 0
6     top = len(li) - 1
7     index = -1
8     while top >= bottom and index == -1:
9         mid = int(math.floor((top + bottom) / 2.0))
10        if li[mid] == element:
11            index = mid
12        elif li[mid] > element:
13            top = mid - 1
14        else:
15            bottom = mid + 1
16
17    return index
18
19
20 li = [2, 5, 7, 9, 11, 17, 222]
21 print(bin_search(li, 11))
22 print(bin_search(li, 12))
```

q058.py

Code python :



```
1 def binary_search(lst, item):
2     low = 0
3     high = len(lst) - 1
4
5     while low <= high:
6         mid = round((low + high) / 2)
7
8         if lst[mid] == item:
9             return mid
10        elif lst[mid] > item:
11            high = mid - 1
12        else:
13            low = mid + 1
14    return None
15
16
17 lst = [
18     1,
19     3,
20     5,
21     7,
22 ]
23 print(binary_search(lst, 9))
```

q058-01.py

### Code python :

```
1 def binary_search_Ascending(array, target):
2     lower = 0
3     upper = len(array)
4     print("Array Length:", upper)
5     while lower < upper:
6         x = (lower + upper) // 2
7         print("Middle Value:", x)
8         value = array[x]
9         if target == value:
10            return x
11        elif target > value:
12            lower = x
13        elif target < value:
14            upper = x
15
16
17 Array = [1, 5, 8, 10, 12, 13, 55, 66, 73, 78, 82, 85, 88, 99]
18 print("The Value Found at Index:", binary_search_Ascending(Array, 82))
```

q058-02.py



Code python :

```
1 idx = 0
2
3
4 def bs(num, num_list):
5     global idx
6     if len(num_list) == 1:
7         if num_list[0] == num:
8             return idx
9         else:
10            return "No exit in the list"
11    elif num in num_list[: len(num_list) // 2]:
12        return bs(num, num_list[: len(num_list) // 2])
13    else:
14        idx += len(num_list) // 2
15    return bs(num, num_list[len(num_list) // 2 :])
16
17
18 print(bs(66, [1, 5, 8, 10, 12, 13, 55, 66, 73, 78, 82, 85, 88, 99,
    ↪ 100]))
```

q058-03.py

### Question 59

Veillez générer un flottant aléatoire où la valeur se situe entre 10 et 100 à l'aide du module math.

**Indices :** Utilisez `random.random ()` pour générer un flottant aléatoire dans `[0,1]`.

Code python :

```
1 import random
2 print(random.random()*100)
```

q059.py

Code python :

```
1 import random
2
3 rand_num = random.uniform(10, 100)
4 print(rand_num)
```

q059-01.py

### Question 60

Veillez écrire un programme pour produire un nombre pair aléatoire entre 0 et 10 inclus en utilisant le module aléatoire et la compréhension de la liste.

**Indices :** Utilisez `random.choice()` à un élément aléatoire d'une liste.



Code python :

```
1 import random
2
3 print(random.choice([i for i in range(11) if i % 2 == 0]))
```

q060.py

Code python :

```
1 import random
2
3 resp = [i for i in range(0, 11, 2)]
4 print(random.choice(resp))
```

q060-01.py

### Question 61

Veillez rédiger un programme pour générer une liste avec 5 nombres aléatoires entre 100 et 200 inclusifs.

**Indices :** Utilisez `random.sample()` pour générer une liste de valeurs aléatoires.

Code python :

```
1 import random
2
3 print(random.sample(range(100), 5))
```

q061.py

### Question 62

Veillez écrire un programme pour générer de manière aléatoire une liste avec 5 nombres, qui sont divisibles par 5 et 7, entre 1 et 1000 inclusifs.

**Indices :** Utilisez `random.sample()` pour générer une liste de valeurs aléatoires.

Code python :

```
1 import random
2
3 print(random.sample([i for i in range(1, 1001) if i % 5 == 0 and i % 7
↳ == 0], 5))
```

q062.py

### Question 63

Veillez écrire un programme pour imprimer au hasard un numéro entier entre 7 et 15 inclusif.

**Indices :** Utilisez `random.randrange()`



Code python :

```
1 import random
2
3 print(random.randrange(7, 16))
```

q063.py

### Question 64

Veillez écrire un programme pour comprimer et décompresser la chaîne "Hello World! Hello World! Hello World! Hello World!".

**Indices :** Utilisez `zlib.compress ()` et `zlib.decompress ()` pour compresser et décompresser une chaîne.

Code python :

```
1 import zlib
2
3 s = b"hello world!hello world!hello world!hello world!"
4 t = zlib.compress(s)
5 print(t)
6 t = zlib.decompress(t)
7 print(t)
```

q064.py

### Question 65

Rédiger un programme pour mélanger et imprimer la liste `[3,6,7,8]`.

**Indices :** Utilisez la fonction `Shuffle()` pour mélanger une liste.

Code python :

```
1 import random
2
3 lst = [3,6,7,8]
4 random.shuffle(lst)
5 print(lst)
```

q065.py

Code python :

```
1 import random
2
3 lst = [3,6,7,8]
4 random.shuffle(lst)
5 print(lst)
```

q065-01.py

**Question 66**

Écrire un programme pour générer toutes les phrases où le sujet est dans ["I", "You"] et le verbe est dans ["Play", "Love"] et l'objet est dans ["Hockey", "Football"].

Code python :

```
1 subjects = ["I", "You"]
2 verbs = ["Play", "Love"]
3 objects = ["Hockey", "Football"]
4 for subject in subjects:
5     for verb in verbs:
6         for obj in objects:
7             sentence = f"{subject} {verb} {obj}."
8             print(sentence)
```

q066.py

Code python :

```
1 import itertools
2
3 subject = ["I", "You"]
4 verb = ["Play", "Love"]
5 objects = ["Hockey", "Football"]
6
7 sentence = [subject, verb, objects]
8 n = list(itertools.product(*sentence))
9 for i in n:
10     print(i)
```

q066-01.py

**Question 67**

En utilisant la compréhension de liste, veuillez écrire un programme pour imprimer la liste après avoir supprimé les nombres divisibles par 5 et 7 dans [12,24,35,70,88,120,155].

Code python :

```
1 li = [12, 24, 35, 70, 88, 120, 155]
2 li = [x for x in li if x % 5 != 0 and x % 7 != 0]
3 print(li)
```

q067.py

**Question 68**

En utilisant la compréhension de liste, écrivez un programme qui génère un tableau 3D 3\*5\*8 dont chaque élément est 0. .



Code python :

```
1 array = [[[0 for col in range(8)] for col in range(5)] for row in  
  ↪ range(3)]  
2 print(array)
```

q068.py

### Question 69

En utilisant la compréhension de liste, veuillez écrire un programme pour imprimer la liste après avoir enlevé la valeur 24 dans [12,24,35,24,88,120,155].

**Indices :** Utilisez la méthode de suppression de la liste pour supprimer une valeur.

Code python :

```
1 li = [12, 24, 35, 24, 88, 120, 155]  
2 li = [x for x in li if x != 24]  
3 print(li)
```

q069.py

Code python :

```
1 li = [12, 24, 35, 24, 88, 120, 155]  
2 li.remove(24) # this will remove only the first occurrence of 24  
3 print(li)
```

q069-01.py

### Question 70

Définissez une classe Personne et ses deux classes enfants : Homme et Femme. Toutes les classes ont une méthode "getGenre" qui peut afficher "Homme" pour la classe Homme et "Femme" pour la classe Femme.

**Indices :** Utilisez la subclass(parentClass) pour définir une classe d'enfants.

Code python :



```
1 class Personne(object):
2     def getGenre(self):
3         return "Unknown"
4
5
6 class Homme(Personne):
7     def getGenre(self):
8         return "Homme"
9
10
11 class Femme(Personne):
12     def getGenre(self):
13         return "Femme"
14
15
16 aHomme = Homme()
17 aFemme = Femme()
18 print(aHomme.getGenre())
19 print(aFemme.getGenre())
```

q070.py

### Code python :

```
1 class Personne(object):
2     def __init__(self):
3         self.genre = "unknown"
4
5     def getGenre(self):
6         print(self.genre)
7
8
9 class Homme(Personne):
10     def __init__(self):
11         self.genre = "Homme"
12
13
14 class Femme(Personne):
15     def __init__(self):
16         self.genre = "Femme"
17
18
19 sharon = Femme()
20 doug = Homme()
21 sharon.getGenre()
22 doug.getGenre()
```

q070-01.py





### Question 71

Veillez écrire un programme qui accepte une chaîne de la console et l'imprimez dans l'ordre inverse.

Exemple :

Si la chaîne suivante est donnée en entrée au programme :

Rise pour voter Sir

Ensuite, la sortie du programme doit être :

riS retov ruop esiR

**Code python :**

```
1 s = input()
2 s = s[::-1]
3 print(s)
```

q071.py

### Question 72

Veillez écrire un programme qui accepte une chaîne de caractères de la console et qui imprime les caractères qui ont des index pairs.

Exemple :

Si la chaîne suivante est donnée en entrée au programme :

H1E2L3L4O5W6O7R8L9D

Ensuite, la sortie du programme doit être :

HELLOWORLD

**Indices :** Utilisez la liste [ : : 2] pour itérer une liste par étape 2.

**Code python :**

```
1 s = input()
2 s = s[::2]
3 print(s)
```

q072.py

**Code python :**



```
1 s = "H1e2l3l4o5w6o7r8l9d"
2 s = [s[i] for i in range(len(s)) if i % 2 == 0]
3 print("".join(s))
```

q072-01.py

Code python :

```
1 s = "H1e2l3l4o5w6o7r8l9d"
2 ns = ""
3 for i in range(len(s)):
4     if i % 2 == 0:
5         ns += s[i]
6 print(ns)
```

q072-02.py

### Question 73

Veuillez écrire un programme qui imprime toutes les permutations de [1,2,3]

**Indices :** Utilisez `itertools.permutations()` pour obtenir des permutations de liste.

Code python :

```
1 import itertools
2
3 print(list(itertools.permutations([1, 2, 3])))
```

q073.py

Code python :

```
1 from itertools import permutations
2
3
4 def permutation_generator(iterable):
5     p = permutations(iterable)
6     for i in p:
7         print(i)
8
9
10 x = [1, 2, 3]
11 permutation_generator(x)
```

q073-01.py

### Question 74

Écrire un programme pour résoudre un casse-tête classique de la Chine ancienne :  
Nous comptons 35 têtes et 94 pattes parmi les poulets et les lapins d'une ferme.  
Combien de lapins et de poulets avons-nous ?



**Indices :** Utilisez pour la boucle pour itérer toutes les solutions possibles.

**Code python :**

```
1 def solve(numheads, numlegs):
2     ns = "No solutions!"
3     for i in range(numheads + 1):
4         j = numheads - i
5         if 2 * i + 4 * j == numlegs:
6             return i, j
7     return ns, ns
8
9
10 numheads = 35
11 numlegs = 94
12 solutions = solve(numheads, numlegs)
13 print(solutions)
```

q074.py

### Question 75

Écrivez une fonction pour calculer 5/0 et utilisez try/except pour attraper les exceptions.

**Code python :**

```
1 def divide():
2     return 5/0
3
4 try:
5     divide()
6 except ZeroDivisionError as ze:
7     print("Why on earth you are dividing a number by ZERO!!")
8 except:
9     print("Any other exception")
```

q117.py

### Question 76

Définir une classe d'exception personnalisée qui prend un message sous forme de chaîne comme attribut.

**Code python :**



```
1
2 class CustomException(Exception):
3     """Exception raised for custom purpose
4
5     Attributes:
6         message -- explanation of the error
7     """
8
9     def __init__(self, message):
10         self.message = message
11
12
13 num = int(input())
14
15 try:
16     if num < 10:
17         raise CustomException("Input is less than 10")
18     elif num > 10:
19         raise CustomException("Input is grater than 10")
20 except CustomException as ce:
21     print("The error raised: " + ce.message)
```

q118.py

### Question 77

Écrire un programme pour calculer  $1/2 + 2/3 + 3/4 + \dots + n/n + 1$  avec une entrée n. Avec la valeur suivante :

5

La sortie sera :

3.55

Code python :

```
1 n = int(input())
2 sum = 0
3 for i in range(1, n + 1):
4     sum += i / (i + 1)
5 print(round(sum, 2)) # rounded to 2 decimal point
```

q119.py

Code python :

```
1 def question_59(n):
2     print(round(sum(map(lambda x: x / (x + 1), range(1, n + 1))), 2))
3
4
5 question_59(5)
```

q119-01.py

**Question 78**

Veillez écrire un programme pour imprimer la liste après avoir enlevé les nombres pairs dans [5,6,77,45,22,12,24].

**Code python :**

```
1 def isEven(n):
2     return n % 2 != 0
3
4
5 li = [5, 6, 77, 45, 22, 12, 24]
6 lst = list(filter(isEven, li))
7 print(lst)
```

q120.py

**Code python :**

```
1 li = [5, 6, 77, 45, 22, 12, 24]
2 lst = list(filter(lambda n: n % 2 != 0, li))
3 print(lst)
```

q120-01.py

**Question 79**

En utilisant la compréhension de liste, veuillez écrire un programme pour imprimer la liste après avoir enlevé les 0ème, 2ème, 4ème, 6ème nombres dans [12,24,35,70,88,120,155].

**Code python :**

```
1 li = [12, 24, 35, 70, 88, 120, 155]
2 li = [li[i] for i in range(len(li)) if i % 2 != 0 and i <= 6]
3 print(li)
```

q121.py

**Code python :**

```
1 """Solution by: popomaticbubble
2 """
3
4 orig_lst = [12, 24, 35, 70, 88, 120, 155]
5 indices = [0, 2, 4, 6]
6
7 new_list = [i for (j, i) in enumerate(orig_lst) if j not in indices]
8 print(new_list)
```

q121-01.py

**Question 80**

En utilisant la compréhension de liste, veuillez écrire un programme pour imprimer la liste après avoir enlevé les 2ème à 4ème nombres dans [12,24,35,70,88,120,155].

Code python :

```
1 # to be written
2 li = [12, 24, 35, 70, 88, 120, 155]
3 li = [li[i] for i in range(len(li)) if i < 2 or i > 4]
4 print(li)
```

q122.py

Code python :

```
1 orig_list = [12, 24, 35, 70, 88, 120, 155]
2 new_list = [i for (j, i) in enumerate(orig_list) if j not in range(2,
↵ 5)]
3 print(new_list)
```

q122-01.py

**Question 81**

En utilisant la compréhension de liste, veuillez écrire un programme pour imprimer la liste après avoir enlevé les 0ème, 4ème et 5ème nombres dans [12,24,35,70,88,120,155].

Code python :

```
1 li = [12, 24, 35, 70, 88, 120, 155]
2 li = [li[i] for i in range(len(li)) if i not in (0, 4, 5)]
3 print(li)
```

q123.py

Code python :

```
1 """Solution by: pratikb0501
2 """
3
4 li = [12, 24, 35, 70, 88, 120, 155]
5 print(list(j for i, j in enumerate(li) if i != 0 and i != 4 and i !=
↵ 5))
```

q123-01.py

**Question 82**

Avec deux listes données [1,3,6,78,35,55] et [12,24,35,24,88,120,155], écrivez un programme pour créer une liste dont les éléments sont l'intersection des listes données ci-dessus.



Code python :

```
1 list1 = [1, 3, 6, 78, 35, 55]
2 list2 = [12, 24, 35, 24, 88, 120, 155]
3 set1 = set(list1)
4 set2 = set(list2)
5 intersection = set1 & set2
6 print(intersection)
```

q124.py

Code python :

```
1 list1 = [1, 3, 6, 78, 35, 55]
2 list2 = [12, 24, 35, 24, 88, 120, 155]
3 set1 = set(list1)
4 set2 = set(list2)
5 intersection = set.intersection(set1, set2)
6 print(intersection)
```

q124-01.py

### Question 83

Avec une liste donnée [12,24,35,24,88,120,155,88,120,155], écrivez un programme pour imprimer cette liste après avoir supprimé toutes les valeurs en double, en conservant l'ordre original.

Code python :

```
1 li = [12, 24, 35, 24, 88, 120, 155, 88, 120, 155]
2 for i in li:
3     if li.count(i) > 1:
4         li.remove(i)
5 print(li)
```

q125.py

Code python :

```
1 def removeDuplicate(li):
2     seen = {} # dictionary
3     for item in li:
4         if item not in seen:
5             seen[item] = True
6             yield item
7
8
9 li = [12, 24, 35, 24, 88, 120, 155, 88, 120, 155]
10 ans = list(removeDuplicate(li))
11 print(ans)
```

q125-01.py

**Question 84**

Veillez écrire un programme qui compte et imprime les numéros de chaque caractère dans une chaîne de caractères saisie par la console. Par exemple, avec l'entrée suivante :

abcdefghijkl

La sortie est :

a,2

b,2

c,2

d,1

e,1

f,1

g,1

**Code python :**

```
1 import string
2
3 s = input()
4 for letter in string.ascii_lowercase:
5     cnt = s.count(letter)
6     if cnt > 0:
7         print("{} , {}".format(letter, cnt))
```

q126.py

**Code python :**

```
1 s = input()
2 for letter in range(ord("a"), ord("z") + 1): # ord() gets the ascii
    ↪ value of a char
3     letter = chr(letter) # chr() gets the char of an ascii value
4     cnt = s.count(letter)
5     if cnt > 0:
6         print(f"{letter} , {cnt}")
```

q126-01.py

**Code python :**

```
1 s = "abcdefghijkl"
2 for i in sorted(set(s)):
3     print(f"{i} , {s.count(i)}")
```

q126-02.py

**Code python :**





```
1 def character_counter(text):
2     characters_list = list(text)
3     char_count = {}
4     for x in characters_list:
5         if x in char_count.keys():
6             char_count[x] += 1
7         else:
8             char_count[x] = 1
9     return char_count
10
11
12 def dict_viewer(dictionary):
13     for x, y in dictionary.items():
14         print(f"{x},{y}")
15
16
17 text = input("> ")
18 dict_viewer(character_counter(text))
```

q126-03.py

### Question 85

A partir de la feuille de résultats des participants à la journée sportive de votre université, vous devez trouver le score du deuxième. On vous donne les scores. Classez-les dans une liste et trouvez le score du deuxième.

Si la chaîne suivante est donnée en entrée au programme :

5

2 3 6 6 5

La sortie est :

5

Code python :

```
1 n = int(input())
2 arr = map(int, input().split())
3 arr = list(set(arr))
4 arr.sort()
5 print(arr[-2])
```

q127.py

Code python :



```
1 num = int(input("Enter num: "))
2 L = []
3
4 while True:
5     L.append(num)
6     num = int(input("Enter another: "))
7     if num == 0:
8         break
9
10 L1 = list(set(L[:]))
11 L2 = sorted(L1)
12 print(L2)
13
14 print(f"The runner up is {L2[-2]}")
```

q127-01.py

### Code python :

```
1 num = int(input())
2 scores = list(map(int, input().split(' ')))
3 winner = max(scores)
4 lst = []
5
6 if len(scores) != num:
7     print('length of score is greater than input given')
8 else:
9     for score in scores:
10         if winner > score:
11             lst.append(score)
12
13 runnerup = max(lst)
14 print(runnerup)
```

q127-02.py

### Question 86

On vous donne une chaîne de caractères S et une largeur W. Votre tâche consiste à envelopper la chaîne de caractères dans un paragraphe de largeur.

Si la chaîne suivante est donnée en entrée au programme :

ABCDEFGHIIJKLIMNOQRSTUVWXYZ

4

La sortie est :

ABCD

EFGH

IJKL

IMNO

QRST



UVWX

YZ

Code python :

```
1 import textwrap
2
3
4 def wrap(string, max_width):
5     string = textwrap.wrap(string, max_width)
6     string = "\n".join(string)
7     return string
8
9
10 if __name__ == "__main__":
11     string, max_width = input(), int(input())
12     result = wrap(string, max_width)
13     print(result)
```

q128.py

Code python :

```
1 import textwrap
2
3 string = input()
4 width = int(input())
5
6 print(textwrap.fill(string, width))
```

q128-01.py

Code python :

```
1 from textwrap import wrap
2
3 x = str(input(": "))
4 w = int(input())
5 z = list(wrap(x, w))
6 for i in z:
7     print(i)
```

q128-02.py

Code python :

```
1 import textwrap
2
3 string = input("")
4 print("\n".join(textwrap.wrap(string, width=int(input("")))))
```

q128-03.py



Code python :

```

1 import itertools
2
3 string = input("> ")
4 width_length = int(input("What is the width of the groupings? "))
5
6
7 def grouper(string, width):
8     iters = [iter(string)] * width
9     return itertools.zip_longest(*iters, fillvalue="")
10
11
12 def displayer(groups):
13     for x in groups:
14         if x == "":
15             continue
16         else:
17             print("".join(x))
18
19
20 displayer(grouper(string, width_length))

```

q128-04.py

### Question 87

On vous donne un nombre entier, N. Votre tâche consiste à imprimer un rangoli alphabétique de taille N. (Le rangoli est une forme d'art populaire indien basé sur la création de motifs).

Différentes tailles de rangoli alphabétique sont présentées ci-dessous :

size 3

```

--c--
-c-b-c-
c-b-a-b-c
-c-b-c-
--c--

```

size 5

```

-----e-----
-----e-d-e-----
---e-d-c-d-e---
-e-d-c-b-c-d-e-
e-d-c-b-a-b-c-d-e
-e-d-c-b-c-d-e-

```



—e-d-c-d-e—  
——e-d-e——  
——e——

Code python :

```
1 import string
2
3
4 def print_rangoli(size):
5     n = size
6     alph = string.ascii_lowercase
7     width = 4 * n - 3
8
9     ans = []
10    for i in range(n):
11        left = "-".join(alph[n - i - 1: n])
12        mid = left[-1:0:-1] + left
13        final = mid.center(width, "-")
14        ans.append(final)
15
16    if len(ans) > 1:
17        for i in ans[n - 2:: -1]:
18            ans.append(i)
19    ans = "\n".join(ans)
20    print(ans)
21
22
23 if __name__ == "__main__":
24     n = int(input())
25     print_rangoli(n)
```

q129.py

Code python :

```
1 def rangoli(n):
2     # your code goes here
3     l1 = list(map(chr, range(97, 123)))
4     x = l1[n - 1 :: -1] + l1[1:n]
5     mid = len("-".join(x))
6     for i in range(1, n):
7         print("-".join(l1[n - 1 : n - i : -1] + l1[n - i :
8             ↪ n])).center(mid, "-")
9     for i in range(n, 0, -1):
10        print("-".join(l1[n - 1 : n - i : -1] + l1[n - i :
11            ↪ n])).center(mid, "-")
12
13 rangoli(5)
```

q129-01.py

**Question 88**

Etant donné 2 ensembles d'entiers, M et N, imprimez leur différence symétrique par ordre croissant. Le terme "différence symétrique" indique les valeurs qui existent dans M ou N mais qui n'existent pas dans les deux.

La première ligne d'entrée contient un entier, M. La deuxième ligne contient M entiers séparés par des espaces. La troisième ligne contient un entier, N. La quatrième ligne contient N entiers séparés par des espaces.

```
4
2 4 5 9
4
2 4 11 12
```

La sortie est :

```
5
9
11
12
```

Code python :

```
1 if __name__ == "__main__":
2     n = int(input())
3     set1 = set(map(int, input().split()))
4
5     m = int(input())
6     set2 = set(map(int, input().split()))
7
8     ans = list(set1 ^ set2)
9     ans.sort()
10    for i in ans:
11        print(i)
```

q130.py

**Question 89**

On vous donne des mots. Certains mots peuvent se répéter. Pour chaque mot, indiquez le nombre d'occurrences. L'ordre de sortie doit correspondre à l'ordre d'apparition du mot en entrée.

Voir l'exemple d'entrée/sortie pour plus de précisions.

```
4
bcdef
abcdefg
bcde
bcdef
```

La sortie est :



3

2 1 1

Code python :

```
1 word = input()
2 dct = {}
3 for i in word:
4     dct[i] = dct.get(i, 0) + 1
5
6 dct = sorted(dct.items(), key=lambda x: (-x[1], x[0]))
7 for i in dct:
8     print(i[0], i[1])
```

q131.py

### Question 90

Votre tâche consiste à compter la fréquence des lettres de la chaîne et à imprimer les lettres par ordre décroissant de fréquence.

Si la chaîne suivante est donnée en entrée du programme :

aabbbccde

La sortie est :

b 3

a 2

c 2

d 1

e 1

Code python :

```
1 word = input()
2 dct = {}
3 for i in word:
4     dct[i] = dct.get(i, 0) + 1
5
6 dct = sorted(dct.items(), key=lambda x: (-x[1], x[0]))
7 for i in dct:
8     print(i[0], i[1])
```

q131.py

Code python :



```
1 X = input()
2 my_set = set(X)
3 arr = []
4 for item in my_set:
5     arr.append([item, X.count(item)])
6 tmp = sorted(arr, key=lambda x: (-x[1], x[0]))
7
8 for i in tmp:
9     print(i[0] + " " + str(i[1]))
```

q131-01.py

**Code python :**

```
1 s = list(input())
2
3 dict_count_ = {k: s.count(k) for k in s}
4 list_of_tuples = [(k, v) for k, v in dict_count_.items()]
5 list_of_tuples.sort(key=lambda x: x[1], reverse=True)
6
7 for item in list_of_tuples:
8     print(item[0], item[1])
```

q131-02.py

**2 Site 2**

Lien vers le site d'origine

**Question 1**

Écrivez une fonction **precedent\_suivant()** qui lit un numéro entier et renvoie ses numéros précédents et suivants.

Exemple d'entrée :

precedent\_suivant(179)

Exemple de sortie :

(178, 180)

**Code python :**

```
1 def previous_next(num):
2     # Your code here
3     return (num - 1, num + 1)
4
5
6 # Invoke the function with any integer as its argument
7 print(previous_next(179))
```

q075.py



**Question 2**

N étudiants prennent K pommes et les distribuent entre eux uniformément. La partie restante (indivisible) reste dans le panier. Combien de pommes aura chaque étudiante et combien resteront dans le panier ?

La fonction lit les nombres n et k et renvoie les deux réponses pour les questions ci-dessus.

Exemple d'entrée :

Apple\_sharing(6, 50)

Exemple de sortie :

(8, 2)

Code python :

```
1 def apple_sharing(n, k):  
2     # Your code here  
3     return (round(k / n), k % n)  
4  
5  
6 print(apple_sharing(6, 50))
```

q076.py

**Question 3**

Écrivez une fonction appelée **carre()** qui calcule la valeur du carré d'un nombre.

Exemple d'entrée :

carre(6)

Exemple de sortie :

36

Code python :

```
1 def square(num):  
2     # Your code here  
3     return num**2  
4  
5  
6 print(square(6))
```

q077.py

**Question 4**

Écrire la fonction **heures\_minutes()** pour transformer le nombre donné en secondes en heures et minutes.

Exemple 1 :



heures\_minutes(3900)

sortie : (1, 5)

Exemple 2 :

heures\_minutes(60)

sortie : (0, 1)

**Code python :**

```
1 def hours_minutes(seconds):
2     # Your code here
3     hours = seconds // 3600
4     remaining_seconds = seconds % 3600
5     minutes = remaining_seconds // 60
6     return (hours, minutes)
7
8
9 # Invoke the function and pass any integer as its argument
10 print(hours_minutes(3900))
11 print(hours_minutes(60))
```

q078.py

### Question 5

Étant donné deux horodatages du même jour. Chaque horodatage est représenté par un nombre :

- d'heures
- de minutes
- de secondes

L'instant du premier horodatage s'est produit avant l'instant du second. Calculez le nombre de secondes qui se sont écoulées entre les deux.

Exemple 1 :

two\_timestamp(1,1,1,2,2,2)

Sortie : 3661

Exemple 2 :

two\_timestamp(1,2,30,1,3,20)

Sortie : 50

**Code python :**



```
1 def two_timestamp(hr1, min1, sec1, hr2, min2, sec2):
2     # Your code here
3     first_hour = hr1 * 3600
4     first_min = min1 * 60
5     final_first = first_hour + first_min + sec1
6     second_hour = hr2 * 3600
7     second_min = min2 * 60
8     final_second = second_hour + second_min + sec2
9
10    return final_second - final_first
11
12
13 # Invoke the function and pass two timestamps(6 integers) as its
   ↪ arguments
14 print(two_timestamp(1, 1, 1, 2, 2, 2))
```

q079.py

### Question 6

Créez une fonction nommée `two_digits()`.

Étant donné un entier à deux chiffres, `two_digits()` renvoie son chiffre gauche (le chiffre des dizaines) puis son chiffre droit (le chiffre des unités).

Exemple d'entrée :

`two_digits(79)`

Exemple de sortie :

`(7, 9)`

**Code python :**



```
1 def two_digits(number):
2     # Your code here
3     aux = str(number)
4     return (int(aux[0]), int(aux[1]))
5
6
7 # Invoke the function with any two digit integer as its argument
8 print(two_digits(79))
9
10
11 """
12 --- SOLUTION 2 ---
13
14 def two_digits(number):
15     tens_digit = number // 10
16     ones_digit = number % 10
17
18     return tens_digit, ones_digit
19
20 print(two_digits(37))
21 """
```

q080.py

### Question 7

Écrire la fonction nommée `swap_digits()`.

Étant donné un entier à deux chiffres, `swap_digits()` échange ses chiffres et imprime le résultat.

Exemple d'entrée :

`swap_digits(79)`

Exemple de sortie :

97

Code python :

```
1 def swap_digits(num):
2     aux = str(num)[1] + str(num)[0]
3     return int(aux)
4
5
6 # Invoke the function with any two-digit integer as its argument
7 print(swap_digits(79))
```

q081.py

### Question 8



Écrire la fonction `last_two_digits()`. Étant donné un entier supérieur à 9, `last_two_digits()` imprime ses deux derniers chiffres.

Exemple d'entrée :

`last_two_digits(1234)`

Exemple de sortie :

34

**Code python :**

```
1 def last_two_digits(num):
2     if num > 9:
3         return int(str(num)[-2:])
4     else:
5         return num
6
7
8 # Invoke the function with any integer greater than 9
9 print(last_two_digits(212))
```

q082.py

### Question 9

Écrire la fonction `tens_digit()`.

Étant donné un entier, `tens_digit()` renvoie son chiffre de dizaines.

Exemple 1 :

`tens_digit(1234)`

Sortie : 3

Exemple 2 :

`tens_digit(179)`

Sortie : 7

**Code python :**

```
1 def tens_digit(num):
2     return (num // 10) % 10
3
4
5 # Invoke the function with any integer
6 print(tens_digit(198))
```

q083.py

### Question 10

Écrire la fonction `digits_sum()`.

Étant donné un numéro à trois chiffres, `digits_sum()` trouve la somme de ses chiffres.

Exemple d'entrée :



digits\_sum(123)

Exemple de sortie :

6

Code python :

```
1 def digits_sum(num):
2     aux = 0
3     for x in str(num):
4         aux = aux + int(x)
5     return aux
6
7
8 # Invoke the function with any three-digit number
9 print(digits_sum(123))
```

q084.py

### Question 11

Écrire la fonction first\_digit(). Étant donné un nombre réel positif, first\_digit() renvoie son premier chiffre (à droite de la virgule).

Exemple d'entrée :

first\_digit(1.79)

Exemple de sortie :

7

Code python :

```
1 import math
2
3
4 def first_digit(num):
5     return int(str(math.floor(num * 10) / 10)[-1])
6
7
8 def first_digit2(num):
9     result = str(num).split(".")
10    return int(result[1][0])
11
12
13 # Invoke the function with a positive real number. ex. 34.33
14 print(first_digit(2.6))
15 print(first_digit(1.79))
16 print(first_digit2(4.2))
17 print(first_digit2(3.14))
```

q085.py

**Question 12**

Une voiture peut parcourir une distance de N kilomètres par jour. Combien de jours lui faudra-t-il pour parcourir un itinéraire d'une longueur de M kilomètres ? Instructions :

Écrire une fonction `car_route()` qui prend deux arguments :

- la distance qu'elle peut parcourir en un jour
- la distance à parcourir

Cette fonction calcule le nombre de jours qu'il faudra pour parcourir cette distance.

Exemple d'entrée :

`car_route(20, 40)`

Exemple de sortie :

2

**Code python :**

```
1 import math
2
3
4 def car_route(n, m):
5     return int(math.ceil(m / n))
6
7
8 # Invoke the function with two integers
9 print(car_route(35, 50))
```

q086.py

**Question 13**

Écrivez une fonction `century()`. Cette dernière prend une année en paramètre sous la forme d'un entier et renvoi le numéro du siècle.

Exemple d'entrée :

`century(2001)`

Exemple de sortie :

21

**Code python :**



```
1 import math
2
3
4 def century(year):
5     if year % 100 == 0:
6         return math.floor(year / 100)
7     else:
8         return math.floor(year / 100 + 1)
9
10
11 # Invoke the function with any given year
12 print(century(2024))
```

q087.py

### Question 14

Un petit gâteau coûte d euros et c centimes. Écrivez une fonction qui détermine le nombre d'euros et de centimes qu'une personne devrait payer pour n petits gâteaux. La fonction reçoit trois nombres : d, c, n et doit renvoyer deux nombres : le coût total en euros et en centimes.

Exemple d'entrée :

total\_cost(15, 22, 4)

Sortie :

(60, 88)

Code python :

```
1 def total_cost(d, c, n):
2     total_cents = (d * 100 + c) * n
3     total_dollars = total_cents // 100
4     remaining_cents = total_cents % 100
5     return total_dollars, remaining_cents
6
7
8 print(total_cost(15, 22, 4))
```

q088.py

### Question 15

Écrire une fonction day\_of\_week(). On lui fourni un entier k compris entre 1 et 365, la fonction day\_of\_week() trouve le numéro du jour de la semaine pour le k-ième jour de l'année, à condition que le 1er janvier de cette année soit un jeudi.

Les jours de la semaine sont numérotés comme :

0 Dimanche

1 Lundi





2 Mardi ...

6 Samedi

Exemple d'entrée :

day\_of\_week(1)

Exemple de sortie :

4

Code python :

```
1 def day_of_week(k):
2     return (3 + k) % 7
3
4
5 # Invoke function day_of_week with an integer between 1 and 365
6 print(day_of_week(125))
```

q089.py

### Question 16

Soit l'entier  $n$  - le nombre de minutes qui se sont écoulées depuis minuit, combien d'heures et de minutes sont affichées sur l'horloge numérique de 24 heures ? Écrivez une fonction `digital_clock()` pour le calculer. La fonction doit afficher deux nombres : le nombre d'heures (entre 0 et 23) et le nombre de minutes (entre 0 et 59).

Exemple d'entrée :

`digital_clock(150)`

Exemple de sortie :

(2, 30)

Code python :

```
1 def digital_clock(n):
2     return ((n // 60), (n % 60))
3
4
5 # Invoke the function with any integer (minutes after midnight)
6 print(digital_clock(150))
```

q090.py

### Question 17

Créez une fonction nommée `factorial()`, qui reçoit un nombre en tant que paramètre et renvoie le factoriel de cette valeur.

Exemple d'entrée :

`factorielle(8)`

Exemple de sortie :

40320



Code python :

```
1 def factorial(x):
2     result = 1
3     for i in range(1, x + 1):
4         result *= i
5     return result
6
7
8 print(factorial(8))
9
10 ### Solution 2 ###
11
12 # import math
13 #
14 # def factorial(x):
15 #     return math.factorial(x)
16 #
17 # print(factorial(8))
```

q091.py

### Question 18

Créez une fonction nommée `racine()`, qui reçoit un nombre en tant que paramètre et renvoie la racine carrée.

Si le nombre résultant a des décimales, veuillez ne garder que les 2 premiers.

Exemple d'entrée :

`racine(50)`

Exemple de sortie :

7.07

Code python :

```
1 import math
2
3
4 def square_root(number):
5     result = round(math.sqrt(number), 2)
6     return result
7
8
9 print(square_root(50))
```

q092.py

### Question 19

Créez une fonction appelée `squares_dictionary()`. La fonction reçoit un nombre `n` et



devrait générer un dictionnaire qui contient des paires de la forme (n : n \* n) pour chaque nombre dans la plage de 1 à n, inclus.

Imprimez le dictionnaire résultant.

Exemple d'entrée :

squares\_dictionary(8)

Exemple de sortie :

{1 : 1, 2 : 4, 3 : 9, 4 : 16, 5 : 25, 6 : 36, 7 : 49, 8 : 64}

Code python :

```
1 def squares_dictionary(n):
2     new_dict = dict()
3     for i in range(1, n + 1):
4         new_dict[i] = i * i
5     return new_dict
6
7
8 print(squares_dictionary(5))
```

q093.py

### Question 20

Créez une fonction appelée list\_and\_tuple (), qui prend une entrée de n nombres renvoie une liste et un tuple de ces nombres sous form de chaîne.

Imprimez la liste et le tuple sur deux lignes.

Exemple d'entrée :

list\_and\_tuple(34,67,55,33,12,98)

Exemple de sortie :

['34', '67', '55', '33', '12', '98'] ('34', '67', '55', '33', '12', '98')

Code python :

```
1 def list_and_tuple(*nums):
2     new_list = [str(num) for num in nums]
3     new_tuple = tuple(new_list)
4
5     return new_list, new_tuple
6
7
8 result_list, result_tuple = list_and_tuple(5, 4, 13, 24, 45)
9 print(result_list)
10 print(result_tuple)
```

q094.py

### Question 21

Définissez une classe appelée InputOutString qui a au moins deux méthodes :



get\_string pour obtenir une chaîne à partir de l'entrée de la console. print\_string pour imprimer la chaîne en majuscule.

Testez les méthodes de votre classe.

**Code python :**

```
1 class InputOutString:
2     def __init__(self):
3         self.input_string = ""
4
5     def get_string(self):
6         self.input_string = input("Enter a string: ")
7
8     def print_string(self):
9         print(self.input_string.upper())
10
11
12 string_object = InputOutString()
13 string_object.get_string()
14 string_object.print_string()
```

q095.py

### Question 22

Écrivez une fonction print\_formula(), avec un paramètre qui calcule et imprime la valeur en fonction de la formule donnée :

$Q = \text{racine carrée de } (2 * c * d) / h$

Voici les valeurs fixes de C et H :

C est de 50.

H est 30.

D serait le paramètre de la fonction.

Exemple d'entrée :

print\_formula(150)

Sortie :

22

**Code python :**

```
1 import math
2
3
4 def print_formula(d):
5     return round(math.sqrt(2 * 50 * d / 30))
6
7
8 print(print_formula(150))
```

q096.py

**Question 23**

Écrivez une fonction `two_dimensional_list()`, qui prend 2 chiffres (x, y) en entrée et génère une liste à 2 dimensions.

La valeur de l'élément dans la ligne i et la colonne j doit être  $i * j$ .

Exemple d'entrée :

`two_dimensional_list(3,5)`

Exemple de sortie :

`[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]`

**Code python :**

```
1 def two_dimensional_list(n_rows, n_columns):
2     dimensions = [int(x) for x in "{},{ {}".format(n_rows,
3         ↪ n_columns).split(",")]
4     row_num = dimensions[0]
5     col_num = dimensions[1]
6     matrix = [[0 for col in range(col_num)] for row in range(row_num)]
7
8     for row in range(row_num):
9         for col in range(col_num):
10             matrix[row][col] = row * col
11
12     return matrix
13
14 print(two_dimensional_list(3, 5))
```

q097.py

**Question 24**

Écrire une fonction `sequence_of_words`, qui accepte en entrée une séquence de mots séparés par des virgules (une chaîne).

Imprimer les mots dans une séquence séparée par des virgules après les avoir triés par ordre alphabétique.

Exemple d'entrée :

`sequence_of_words("sans, bonjour, sac, monde")`

Exemple de sortie :

Sac, bonjour, sans, monde

**Code python :**



```
1 def sequence_of_words(words):
2     items = [x for x in "{}".format(words).split(",")]
3     items.sort()
4     return ",".join(items)
5
6
7 print(sequence_of_words("this,is,sorted"))
```

q098.py

### Question 25

Écrire une fonction appelée `remove_duplicate_words()` qui accepte en entrée une séquence de mots séparés par des espaces et qui renvoie les mots après avoir supprimé tous les mots en double et les avoir triés par ordre alphanumérique.

Exemple d'entrée :

`remove_duplicate_words("Hello World and Practice rend à nouveau parfait et bonjour le monde")`

Exemple de sortie :

Encore une fois et bonjour fait un monde de pratique parfait

Code python :

```
1 def remove_duplicate_words(text):
2     words = text.split()
3     return " ".join(sorted(list(set(words))))
4
5
6 print(
7     remove_duplicate_words(
8         "hello world and practice makes perfect and hello world again"
9     )
10 )
```

q099.py

### Question 26

Écrire une fonction `divisible_binary()` qui prend en entrée une séquence de nombres binaires à 4 chiffres séparés par des virgules et vérifie s'ils sont divisibles par 5. Imprimer les nombres qui sont divisibles par 5 dans une séquence séparée par des virgules.

Exemple d'entrée :

`divisible_binary("1000,1100,1010,1111")`

Exemple de sortie :

1010,1111



Code python :

```
1 def divisible_binary(binary_sequence):
2     divisible_numbers = []
3     binary_numbers = [x for x in binary_sequence.split(",")]
4     for binary_num in binary_numbers:
5         int_binary_num = int(binary_num, 2)
6         if not int_binary_num % 5:
7             divisible_numbers.append(binary_num)
8
9     return ",".join(divisible_numbers)
10
11
12 print(divisible_binary("1000,1100,1010,1111"))
```

q100.py

### Question 27

Définir une fonction nommée `all_digits_even()` pour identifier et imprimer tous les nombres entre 1000 et 3000 (inclus) où chaque chiffre du nombre est un nombre pair. Affichez les nombres résultants dans une séquence séparée par des virgules sur une seule ligne.

Code python :

```
1 def all_digits_even():
2     values = []
3     for i in range(1000, 3001):
4         s = str(i)
5         if (
6             (int(s[0]) % 2 == 0)
7             and (int(s[1]) % 2 == 0)
8             and (int(s[2]) % 2 == 0)
9             and (int(s[3]) % 2 == 0)
10        ):
11            values.append(s)
12
13    return ",".join(values)
14
15
16 print(all_digits_even())
```

q101.py

### Question 28

Écrire une fonction nommée `letters_and_digits()` qui prend une phrase en entrée et calcule le nombre de lettres et de chiffres qu'elle contient.

Exemple d'entrée :



```
letters_and_digits("Hello World! 123")
```

Exemple de sortie :

Lettres 10 Chiffres 3

**Code python :**

```
1 def letters_and_digits(text):
2     counts = {"DIGITS": 0, "LETTERS": 0}
3     for char in text:
4         if char.isdigit():
5             counts["DIGITS"] += 1
6         elif char.isalpha():
7             counts["LETTERS"] += 1
8         else:
9             pass
10
11     return f"Lettres {counts['LETTERS']} \nChiffres {counts['DIGITS']}"
12
13
14 print(letters_and_digits("hello world! 123"))
```

q102.py

### Question 29

Écrivez un programme `number_of_uppercase()` qui accepte une phrase et calcule le nombre de lettres majuscules et minuscules.

Exemple d'entrée :

```
number_of_uppercase("Hello World!")
```

Exemple de sortie :

Majuscule 1 Minuscule 9

**Code python :**





```
1 # Your code here
2 def number_of_uppercase(string):
3     counts = {"UPPERCASE": 0, "LOWERCASE": 0}
4     for char in string:
5         if char.isupper():
6             counts["UPPERCASE"] += 1
7         elif char.islower():
8             counts["LOWERCASE"] += 1
9         else:
10            pass
11
12    return f"Majuscule {counts['UPPERCASE']} \nMinuscule
13           ↪ {counts['LOWERCASE']}"
14
15 print(number_of_uppercase("Hello world!"))
```

q103.py

### Question 30

Écrivez un programme `computed_value()` pour calculer la somme d'un + aa + aaa + aaaa, où «a» est un chiffre donné.

Exemple d'entrée :

`computed_value(9)`

Exemple de sortie :

11106

Code python :

```
1 def computed_value(param):
2     result = 0
3     for i in range(1, 5):
4         concatenated_number = int(str(param) * i)
5         result += concatenated_number
6     return result
7
8
9 print(computed_value(9))
```

q104.py

### Question 31

Écrivez une fonction nommée `square_odd_numbers()` qui accepte en entrée une chaîne de nombres séparés par des virgules, ne met au carré que les nombres impairs et renvoie les résultats sous la forme d'une liste.

Exemple d'entrée :



```
square_odd_numbers("1,2,3,4,5,6,7,8,9")
```

Exemple de sortie :

```
[1, 9, 25, 49, 81]
```

Code python :

```
1 def square_odd_numbers(numbers_str):
2     numbers_list = numbers_str.split(",")
3     squared_odd_numbers = []
4
5     for num_str in numbers_list:
6         if num_str.isdigit():
7             num = int(num_str)
8
9             if num % 2 != 0:
10                squared_odd_numbers.append(num**2)
11
12    return squared_odd_numbers
13
14
15 print(square_odd_numbers("1,2,3,4,5,6,7"))
16
17
18 ### SOLUTION 2 ### (List Comprehension)
19
20 # def square_odd_numbers(numbers):
21 #     number_list = [int(num) for num in numbers.split(',')]
22 #     squared_odd_numbers = [num**2 for num in number_list if num % 2
23 ↪    != 0]
24 #     return squared_odd_numbers
25
26 # print(square_odd_numbers("1,2,3,4,5,6,7"))
```

q105.py

### Question 32

Écrire une fonction nommée `net_amount()` qui calcule le montant net d'un compte bancaire sur la base d'un journal de transactions provenant de l'entrée. Le format du journal des transactions est le suivant :

D 100

W 200

D signifie dépôt tandis que w signifie le retrait.

Exemple d'entrée :

```
net_amount("D 300 D 300 W 200 D 100")
```

Exemple de sortie :

```
500
```



Code python :

```
1 def net_amount(param):
2     total = 0
3     values = param.split()
4     for x in range(len(values)):
5         if values[x] == "D":
6             total += int(values[x + 1])
7         elif values[x] == "W":
8             total -= int(values[x + 1])
9     return total
10
11
12 print(net_amount("D 300 W 200 D 400"))
```

q106.py

### Question 33

Un site Web oblige les utilisateurs à saisir un nom d'utilisateur et un mot de passe pour s'inscrire. Écrivez une fonction nommée `valid_password()` pour vérifier la validité de l'entrée de mot de passe par les utilisateurs. Voici les critères de vérification du mot de passe :

- Au moins 1 lettre entre [A-Z].
- Au moins 1 nombre entre [0-9].
- Au moins 1 lettre entre [A-Z].
- Au moins 1 caractère de [\$ # @].
- Longueur minimale du mot de passe : 6.
- Longueur maximale du mot de passe : 12.

Votre programme doit accepter un mot de passe et le vérifier en fonction des critères précédents. Si le mot de passe est validé avec succès, la fonction renvoie la chaîne suivante "Mot de passe valide". Sinon, il renvoie "mot de passe non valide. Veuillez réessayer". Exemple d'entrée :

```
valid_password("ABD1234 @ 1")
```

Exemple de sortie :

"Mot de passe valide"

Code python :



```
1 import re
2
3
4 def valid_password(password):
5     pattern =
6         ↪ re.compile(r"^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[$#@]).{6,12}$")
7
8     if not pattern.match(password):
9         return "Invalid password. Please try again"
10    else:
11        return "Valid password"
12
13 print(valid_password("ABd1234@1"))
```

q107.py

### Question 34

Écrivez une fonction `sort_tuples_ascending()` pour trier les tuples (nom, âge, score) par ordre croissant, où nom, âge et score sont tous des chaînes de caractères. Les critères de tri sont :

- Trier basé sur le nom.
- Puis trier en fonction de l'âge.
- Puis trier par score.

La priorité est le nom > Age > Score.

Exemple d'entrée :

```
sort_tuples_ascending([«Tom, 19,80», «John, 20,90», «Jony, 17,91», «Jony, 17,93», «Jason, 21,85»])
```

Exemple de sortie :

```
[('Jason', '21', '85'), ('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Tom', '19', '80')]
```

**Code python :**



```
1 from operator import itemgetter
2
3
4 def sort_tuples_ascending(data):
5     tuples_list = [tuple(entry.split(",")) for entry in data]
6
7     sorted_tuples = sorted(tuples_list, key=itemgetter(0, 1, 2))
8
9     return sorted_tuples
10
11
12 example_input = ["Tom,19,80", "John,20,90", "Jony,17,91", "Jony,17,93",
13 ↪ "Jason,21,85"]
14
15 result = sort_tuples_ascending(example_input)
16 print(result)
```

q108.py

### Question 35

Définir une classe avec une fonction génératrice qui peut itérer les nombres qui sont divisibles par 7 entre un intervalle donné 0 et n.

Code python :

```
1 class DivisibleBySevenIterator:
2     def __init__(self, n):
3         self.n = n
4
5     def generate_divisible_by_seven(self):
6         for number in range(self.n + 1):
7             if number % 7 == 0:
8                 yield number
9
10
11 n_value = 50
12 divisible_by_seven_iterator = DivisibleBySevenIterator(n_value)
13
14 for num in divisible_by_seven_iterator.generate_divisible_by_seven():
15     print(num)
```

q109.py

### Question 36

Un robot se déplace dans un plan à partir du point d'origine (0,0). Le robot peut se déplacer vers le HAUT, le BAS, la GAUCHE et la DROITE avec des étapes données. La trace du mouvement du robot est présentée sous la forme d'une liste comme la



suivante :

```
["UP 5", "DOWN 3", "LEFT 3", "RIGHT 2"]
```

Les nombres qui suivent la direction sont des pas. Veuillez écrire un programme nommé `compute_robot_distance()` pour calculer la distance finale après une séquence de mouvements à partir du point d'origine. Si la distance est un flottant, il suffit d'imprimer l'entier le plus proche. Exemple d'entrée :

```
compute_robot_distance(["UP 5", "DOWN 3", "LEFT 3", "RIGHT 2"])
```

Exemple de sortie :

2

Code python :

```
1 def compute_robot_distance(movements):
2     x, y = 0, 0
3
4     for move in movements:
5         direction, steps = move.split()
6         steps = int(steps)
7
8         if direction == "UP":
9             y += steps
10        elif direction == "DOWN":
11            y -= steps
12        elif direction == "LEFT":
13            x -= steps
14        elif direction == "RIGHT":
15            x += steps
16
17        distance = (x**2 + y**2) ** 0.5
18        rounded_distance = round(distance)
19
20    return rounded_distance
21
22
23 print(compute_robot_distance(["UP 5", "DOWN 3", "LEFT 3", "RIGHT 2"]))
```

q110.py

### Question 37

Écrivez une fonction appelée `compute_word_frequency()` pour calculer la fréquence des mots à partir d'une chaîne de caractères.

- Placez chaque mot séparé par un espace dans un dictionnaire et comptez sa fréquence.
- Classez le dictionnaire par ordre alphanumérique et imprimez dans la console chaque clé sur une nouvelle ligne.

Exemple d'entrée :



```
compute_word_frequency("New to Python or choosing between Python 2 and Python 3 ? Read Python 2 or Python 3.")
```

Exemple de sortie :

```
2 : 2
3. : 1
3? : 1
New : 1
Python : 5
Read : 1
and : 1
between : 1
choosing : 1
or : 2
to : 1
```

Code python :

```
1 def compute_word_frequency(sentence):
2     words = sentence.split()
3
4     word_frequency = {}
5
6     for word in words:
7         word_frequency[word] = word_frequency.get(word, 0) + 1
8
9     sorted_word_frequency = sorted(word_frequency.items(), key=lambda
    ↪ x: x[0])
10
11    for word, frequency in sorted_word_frequency:
12        print(f"{word}: {frequency}")
13
14
15 input_sentence = "New to Python or choosing between Python 2 and Python
    ↪ 3? Read Python 2 or Python 3."
16 compute_word_frequency(input_sentence)
```

q111.py

### Question 38

En Python, une classe est une structure qui permet d'organiser et d'encapsuler des données et des fonctionnalités connexes. Les classes sont une caractéristique fondamentale de la programmation orientée objet (POO), un paradigme de programmation qui utilise des objets pour modéliser et organiser le code.

En termes simples, une classe est comme un plan ou un modèle pour créer des objets. Un objet est une instance spécifique d'une classe à laquelle sont associés des attributs (données) et des méthodes (fonctions). Les attributs représentent les caractéristiques de l'objet et les méthodes représentent les actions que l'objet peut effectuer. Exemple :



```
1 class Student:
2     def __init__(self, name, age, grade): # These are its attributes
3         self.name = name
4         self.age = age
5         self.grade = grade
6
7     def introduce(self): # This is a method
8         return f"Hello! I am {self.name}, I am {self.age} years old,
9             ↪ and my current grade is {self.grade}."
10
11     def study(self, hours): # This is another method
12         self.grade += hours * 0.5
13         return f"After studying for {hours} hours, {self.name}'s new
14             ↪ grade is {self.grade}."
15
16 student1 = Student("Ana", 20, 80)
17 print(student1.introduce())
18 print(student1.study(3))
```

q112depart.py

Dans ce code :

- La classe Student possède une méthode `__init__` pour initialiser les attributs nom, âge et classe de l'étudiant.
- `introduce` est une méthode qui imprime un message de présentation de l'étudiant.
- `study` est une méthode qui simule l'acte d'étudier et met à jour la note de l'étudiant.

Instructions :

Pour réaliser cet exercice, copiez le code fourni dans l'exemple et collez-le dans votre fichier. Exécutez le code et testez sa fonctionnalité. Essayez de modifier différents aspects du code pour observer son comportement. Cette approche pratique vous aidera à comprendre la structure et le comportement de la classe Étudiant. Une fois que vous serez familiarisé avec le code et ses effets, n'hésitez pas à passer à l'exercice suivant.

**Code python :**





```
1 class Student:
2     def __init__(self, name, age, grade): # These are its attributes
3         self.name = name
4         self.age = age
5         self.grade = grade
6
7     def introduce(self): # This is a method
8         return f"Hello! I am {self.name}, I am {self.age} years old,
9             ↪ and my current grade is {self.grade}."
10
11     def study(self, hours): # This is another method
12         self.grade += hours * 0.5
13         return f"After studying for {hours} hours, {self.name}'s new
14             ↪ grade is {self.grade}."
15
16
17 student1 = Student("Ana", 20, 80)
18
19 print(student1.introduce())
20 print(student1.study(3))
```

q112.py

### Question 39

#### Méthodes `__init__` et `__str__`

En général, lorsque vous travaillez avec des classes, vous rencontrez des méthodes de la forme `__<méthode>__`; ces méthodes sont appelées "méthodes magiques". Il en existe un grand nombre, chacune ayant un objectif spécifique. Cette fois-ci, nous nous concentrerons sur l'apprentissage de deux des méthodes les plus fondamentales. La méthode magique `__init__` est essentielle pour l'initialisation des objets au sein d'une classe. Elle est automatiquement exécutée lorsqu'une nouvelle instance de la classe est créée, ce qui permet d'attribuer des valeurs initiales aux attributs de l'objet. La méthode `__str__` est utilisée pour fournir une représentation sous forme de chaîne de caractères lisible de l'instance, ce qui permet de personnaliser la sortie lors de l'impression de l'objet. Cette méthode est particulièrement utile pour améliorer la lisibilité du code et faciliter le débogage, car elle définit une version conviviale des informations contenues dans l'objet.

Exemple :



```
1 class Person:
2     def __init__(self, name, age, gender):
3         self.name = name
4         self.age = age
5         self.gender = gender
6
7     def __str__(self):
8         return f"{self.name}, {self.age} years old, {self.gender}"
9
10
11 # Create an instance of the Person class
12 person1 = Person("Juan", 25, "Male")
13
14 # Print the information of the person using the __str__ method
15 print(person1) # Output: Juan, 25 years old, Male
```

q113depart.py

Instructions :

- Créez une classe appelée Book qui possède les méthodes `__init__` et `__str__`.
- La méthode `__init__` doit initialiser les attributs `title`, `author` et `year`.
- La méthode `__str__` doit renvoyer une chaîne de caractères représentant les informations d'une instance du livre suivant de cette manière :  
book1 = ("The Great Gatsby", "F. Scott Fitzgerald", 1925)  
print(book1)

```
# Sortie :
#
# Title : Le Grand Gatsby
# Author : F. Scott Fitzgerald
# Year : 1925
```

Code python :



```
1 class Book:
2     def __init__(self, title, author, year):
3         self.title = title
4         self.author = author
5         self.year = year
6
7     def __str__(self):
8         return f"Book Title: {self.title}\nAuthor: {self.author}\nYear:
9             ↪ {self.year}"
10
11 book1 = Book("The Great Gatsby", "F. Scott Fitzgerald", 1925)
12
13 print(book1)
```

q113.py

### Question 40

#### Héritage et polymorphisme

Maintenant que nous avons compris ce qu'est une classe et certaines de ses caractéristiques, abordons deux nouveaux concepts liés aux classes : l'héritage et le polymorphisme. Prenons l'exemple suivant :

```
1 class HighSchoolStudent(Student): # Add the parent class inside the
2     ↪ parenthesis
3     def __init__(self, name, age, grade, specialization):
4         super().__init__(name, age, grade)
5         self.specialization = specialization
6
7     def study(self, hours):
8         return f"{self.name} is a high school student specializing in
9             ↪ {self.specialization} and is studying for {hours} hours for
10             ↪ exams."
11
12 # Creating an instance of HighSchoolStudent
13 high_school_student = HighSchoolStudent("John", 16, 85, "Science")
14 print(high_school_student.introduce()) # We can call this method
15 ↪ thanks to inheritance
16 print(
17     high_school_student.study(4)
18 ) # This method has been slightly modified and now it returns a
19 ↪ different string
```

q114depart.py

En supposant que la classe Student de l'exercice précédent soit codée juste au-dessus de la classe HighSchoolStudent, pour hériter de ses méthodes et attributs, il suffit



d'inclure le nom de la classe dont nous voulons hériter (la classe mère) entre les parenthèses de la classe enfant (HighSchoolStudent). Comme vous pouvez le constater, nous pouvons maintenant utiliser la méthode `introduce` de la classe `Student` sans avoir à la coder à nouveau, ce qui rend notre code plus efficace. Il en va de même pour les attributs ; nous n'avons pas besoin de les redéfinir.

En outre, nous avons la possibilité d'ajouter de nouvelles méthodes exclusivement pour cette classe ou même de remplacer une méthode héritée si nécessaire, comme le montre la méthode `study`, qui est légèrement modifiée à partir de la méthode `Student` ; c'est ce qu'on appelle le polymorphisme.

### Instructions :

- Créez une classe appelée `CollegeStudent` qui hérite de la classe `Student` déjà définie.
- Ajoutez un nouvel attribut appelé `major` pour représenter la spécialité étudiée.
- Modifiez la méthode `introduce` héritée pour qu'elle renvoie cette chaîne de caractères :  
"Bonjour ! Je m'appelle <nom> et je suis étudiant en <major>."
- Ajoutez une nouvelle méthode appelée `attend_lecture` qui renvoie la chaîne suivante :  
"<nom> assiste à une conférence pour les étudiants de <major>".
- Créez une instance de votre nouvelle classe et appelez chacune de ses méthodes.
- Exécutez votre code pour vous assurer qu'il fonctionne.

### Code python :



```
1 class Student:
2     def __init__(self, name, age, grade):
3         self.name = name
4         self.age = age
5         self.grade = grade
6
7     def introduce(self):
8         return f"Hello! I am {self.name}, I am {self.age} years old,
9             ↪ and my current grade is {self.grade}."
10
11     def study(self, hours):
12         return f"{self.name} is studying for {hours} hours."
13
14 class CollegeStudent(Student):
15     def __init__(self, name, age, grade, major):
16         super().__init__(name, age, grade)
17         self.major = major
18
19     def introduce(self):
20         return f"Hi there! I'm {self.name}, a college student majoring
21             ↪ in {self.major}."
22
23     def attend_lecture(self):
24         return f"{self.name} is attending a lecture for {self.major}
25             ↪ students."
26
27 college_student = CollegeStudent("Alice", 20, 90, "Computer Science")
28 print(college_student.introduce())
29 print(college_student.study(3))
30 print(college_student.attend_lecture())
```

q114.py

### Question 41

#### Méthodes statiques

Une méthode statique en Python est une méthode liée à une classe plutôt qu'à une instance de la classe. Contrairement aux méthodes ordinaires, les méthodes statiques n'ont pas accès à l'instance ou à la classe elle-même.

Les méthodes statiques sont souvent utilisées lorsqu'une méthode particulière ne dépend pas de l'état de l'instance ou de la classe. Elles ressemblent davantage à des fonctions utilitaires associées à une classe.



```
1 class Person:
2
3     def __init__(self, name, age):
4         self.name = name
5         self.age = age
6
7     @staticmethod
8     def is_adult(age):
9         return age >= 18
10
11
12 # Creating instances of Person
13 person1 = Person("Alice", 25)
14 person2 = Person("Bob", 16)
15
16 # Using the static method to check if a person is an adult
17 is_adult_person1 = Person.is_adult(person1.age)
18 is_adult_person2 = Person.is_adult(person2.age)
19 print(f"{person1.name} is an adult: {is_adult_person1}")
20 print(f"{person2.name} is an adult: {is_adult_person2}")
```

q115depart.py

Dans cet exemple :

La méthode statique `is_adult` vérifie si une personne est un adulte en fonction de son âge. Elle n'a pas accès directement aux variables d'instance ou de classe.

### Instructions :

- Créez une classe appelée `MathOperations`.
- Créez une méthode statique nommée `add_numbers` qui prend deux nombres en paramètre et renvoie leur somme.
- Créez une instance de la classe `MathOperations`.
- Utilisez la méthode statique `add_numbers` pour additionner deux nombres, par exemple 10 et 15.
- Imprimez le résultat.

Exemple d'entrée :

```
math_operations_instance = MathOperations()
sum_of_numbers = MathOperations.add_numbers(10, 15)
```

Sortie :

25

Code python :



```
1 class MathOperations:
2
3     @staticmethod
4     def add_numbers(num1, num2):
5         return num1 + num2
6
7
8 # You can call the static method without creating an instance
9 sum_of_numbers = MathOperations.add_numbers(10, 15)
10
11 print(f"Sum of Numbers: {sum_of_numbers}")
```

q115.py

### Question 42

#### Méthodes de classe

Une méthode de classe est une méthode liée à la classe et non à l'instance de la classe. Elle prend comme premier paramètre la classe elle-même, souvent nommée "cls". Les méthodes de classe sont définies à l'aide du décorateur @classmethod.

La principale caractéristique d'une méthode de classe est qu'elle peut accéder et modifier les attributs au niveau de la classe, mais qu'elle ne peut pas accéder ou modifier les attributs spécifiques à l'instance puisqu'elle n'a pas accès à une instance de la classe. Les méthodes de classe sont souvent utilisées pour des tâches qui impliquent la classe elle-même plutôt que des instances individuelles.



```
1 class Person:
2     total_people = 0 # Class variable to keep track of the total
3     ↪ number of people
4
5     def __init__(self, name, age):
6         self.name = name
7         self.age = age
8         Person.total_people += (
9             1 # Increment the total_people count for each new instance
10        )
11
12    @classmethod
13    def get_total_people(cls):
14        return cls.total_people
15
16 # Creating instances of Person
17 person1 = Person("Alice", 25)
18 person2 = Person("Bob", 16)
19
20 # Using the class method to get the total number of people
21 total_people = Person.get_total_people()
22 print(f"Total People: {total_people}")
```

q116depart.py

Dans cet exemple :

La méthode de classe `get_total_people` renvoie le nombre total de personnes créées (instances de la classe `Person`).

### Instructions :

- Créez une classe appelée `MathOperations`.
- Dans cette classe, définissez les éléments suivants :
  - Une variable de classe nommée `pi` avec une valeur de 3,14159.
  - Une méthode de classe nommée `calculate_circle_area` qui prend un rayon comme paramètre et renvoie l'aire d'un cercle à l'aide de la formule :  $aire = \pi \text{rayon}^2$ .
- Utilisez la méthode de classe `calculate_circle_area` pour calculer l'aire d'un cercle de rayon 5.
- Imprimez le résultat. (Il n'est pas nécessaire de créer une instance)

Exemple d'entrée :

```
circle_area = MathOperations.calculate_circle_area(5)
```

Sortie :

78.53975

Code python :





```
1 class MathOperations:
2     pi = 3.14159
3
4     @classmethod
5     def calculate_circle_area(cls, radius):
6         area = cls.pi * radius**2
7         return area
8
9
10 circle_area = MathOperations.calculate_circle_area(5)
11
12 print(f"Circle Area: {circle_area}")
```

q116.py

### 3 Site 3

Lien vers le site d'origine

#### Question 1

Définir une fonction nommée `is_two`. Elle doit accepter une entrée et retourner `True` si l'entrée passée est soit le nombre, soit la chaîne `2`, `False` sinon.

Code python :

```
1 def is_two(x):
2     if x == 2 or x == "2":
3         return True
4     else:
5         return False
6
7
8 print(is_two(2))
9 print(is_two("2"))
10 print(is_two(3))
11 print(is_two("3"))
```

q200.py

#### Question 2

Définir une fonction nommée `is_vowel`. Elle doit renvoyer `True` si la chaîne passée est une voyelle, `False` sinon.

Code python :



```
1 def is_vowel(vowel):
2     vowels = "aeiouAEIOU"
3     if vowel in vowels:
4         return True
5     else:
6         return False
7
8
9 print(is_vowel("a"))
10 print(is_vowel("b"))
11 print(is_vowel("A"))
12 print(is_vowel("B"))
```

q201.py

### Question 3

Définir une fonction nommée `is_consonant`. Elle doit retourner `True` si la chaîne passée est une consonne, `False` sinon. Utilisez votre fonction `is_vowel`

Code python :

```
1 def is_consonant(consonant):
2     vowels = "aeiouAEIOU"
3     if consonant not in vowels:
4         return True
5     else:
6         return False
7
8 print(is_consonant("a"))
9 print(is_consonant("b"))
10 print(is_consonant("A"))
11 print(is_consonant("B"))
```

q202.py

### Question 4

Définir une fonction qui accepte une chaîne de caractères qui est un mot. La fonction doit mettre en majuscule la première lettre du mot si celui-ci commence par une consonne.

Code python :



```
1 def cap_if_consonant(x):
2     vowels = "aeiouAEIOU"
3     for letter in vowels:
4         if x[0] in vowels:
5             return x
6         else:
7             return x.capitalize()
8
9
10 print(cap_if_consonant("great job!"))
```

q203.py

### Question 5

Définissez une fonction nommée `calculate_tip`. Elle doit accepter un pourcentage de pourboire (un nombre entre 0 et 1) et le total de l'addition, et renvoyer le montant du pourboire.

Code python :

```
1 def calculate_tip(x, y):
2     bill_total = x
3     tip_percentage = y
4     tip = bill_total * tip_percentage
5     return f"${tip}"
6
7
8 print(calculate_tip(10, 0.15))
```

q204.py

### Question 6

Définissez une fonction nommée `apply_discount`. Elle doit accepter un prix d'origine et un pourcentage de remise, et renvoyer le prix après la remise.

Code python :

```
1 def apply_discount(x, y):
2     original_price = x
3     discount_percentage = y
4     total = original_price - (original_price * discount_percentage)
5     return f"${total}"
6
7
8 print(apply_discount(10, 0.2))
```

q205.py

**Question 7**

Définissez une fonction nommée `handle_commas`. Elle doit accepter en entrée une chaîne de caractères qui est un nombre contenant des virgules, et retourner un nombre en sortie.

Code python :

```
1 def handle_commas(x):
2     no_commas = int(x.replace(",", ""))
3     return no_commas
4
5
6 print(handle_commas("1,000,000"))
```

q206.py

**Question 8**

Définissez une fonction nommée `get_letter_grade`. Elle doit accepter un nombre et retourner la lettre associée à ce nombre (A-F).

Code python :

```
1 def get_letter_grade(grade):
2     if grade >= 100:
3         print("Please enter a grade between 0-100")
4     elif grade >= 90:
5         print(f"{grade} is an A")
6     elif grade >= 80:
7         print(f"{grade} is a B")
8     elif grade >= 70:
9         print(f"{grade} is a C")
10    elif grade >= 60:
11        print(f"{grade} is a D")
12    elif grade <= 59 and grade >= 0:
13        print(f"{grade} is an F")
14    else:
15        print("No negative numbers")
16
17
18 get_letter_grade(95)
19 get_letter_grade(85)
20 get_letter_grade(75)
21 get_letter_grade(65)
22 get_letter_grade(55)
23 get_letter_grade(0)
24 get_letter_grade(105)
25 get_letter_grade(-5)
```

q207.py

**Question 9**

Définissez une fonction nommée `remove_vowels` qui accepte une chaîne et renvoie une chaîne dont toutes les voyelles ont été supprimées.

Code python :

```
1 def remove_vowels(string):
2     vowels = ("a", "e", "i", "o", "u")
3     for x in string.lower():
4         if x in vowels:
5             string = string.replace(x, "")
6     return string
7
8
9 print(remove_vowels("Hello Codeup!"))
```

q208.py

**Question 10**

Définissez une fonction nommée `normalize_name`. Elle doit accepter une chaîne et retourner un identifiant python valide, c'est-à-dire :

- tout ce qui n'est pas un identifiant python valide doit être supprimé
- les espaces blancs de début et de fin doivent être supprimés
- tout doit être en minuscules
- les espaces doivent être remplacés par des traits de soulignement

par exemple :

- Nom deviendra nom
- Prénom deviendra prénom
- Completed deviendra completed

Code python :



```
1 def normalize_name(text):
2     new_string = text.strip("0123456789 ").lower().replace(" ", "_")
3     output = ""
4     if new_string.isidentifier():
5         print(new_string, "is a valid identifier")
6         for i in new_string:
7             if i not in "!@#%$^&*()+=-[]{}\\|?<.>`~":
8                 output += i
9     else:
10        print(new_string, "is NOT a valid identifier")
11    return print(output)
12
13
14 normalize_name("one+one")
15 normalize_name(" Dani Bojado ")
```

q209.py

### Question 11

Écrivez une fonction nommée `cumulative_sum` qui accepte une liste de nombres et renvoie une liste qui est la somme cumulative des nombres de la liste.

- `cumulative_sum([1, 1, 1])` renvoie `[1, 2, 3]`
- `cumulative_sum([1, 2, 3, 4])` renvoie `[1, 3, 6, 10]`

Code python :

```
1 def cumulative_sum(num_list):
2     return [sum(num_list[: i + 1]) for i in range(len(num_list))]
3
4
5 print(cumulative_sum([1, 1, 1]))
6 print(cumulative_sum([1, 2, 3, 4]))
```

q210.py

### Question 12

Soit les deux listes suivantes :

`fruits = ['mango', 'kiwi', 'strawberry', 'guava', 'pineapple', 'mandarin orange']`

`numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4, -2, 5, -9]`

Réécrire l'exemple de code ci-dessus en utilisant la syntaxe de compréhension de liste. Créez une variable nommée `uppercased_fruits` pour contenir la sortie de la compréhension de liste. La sortie devrait être `['MANGO', 'KIWI', etc...]`.

Code python :



```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 uppercased_fruits = [fruit.upper() for fruit in fruits]  
5 print(uppercased_fruits)
```

q211.py

### Question 13

Créer une variable nommée `capitalized_fruits` et utiliser la syntaxe de compréhension de liste pour produire des résultats comme ['Mango', 'Kiwi', 'Strawberry', etc...].

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 capitalized_fruits = [fruit.capitalize() for fruit in fruits]  
5 capitalized_fruits
```

q212.py

### Question 14

Utilisez une compréhension de liste pour créer une variable nommée `fruits_avec_plus_de_deux_vo`.  
Astuce : Vous aurez besoin d'un moyen de vérifier si quelque chose est une voyelle.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 fruits_with_more_than_two_vowels = [fruit for fruit in fruits if (  
5     fruit.count("a") +  
6     fruit.count("e") +  
7     fruit.count("i") +  
8     fruit.count("o") +  
9     fruit.count("u")) > 2]  
10  
11 print(fruits_with_more_than_two_vowels)
```

q213.py

**Question 15**

Créer une variable nommée `fruits_avec_seulement_deux_voyelles`.  
Le résultat devrait être `['mangue', 'kiwi', 'fraise']`.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4 fruits_with_only_two_vowels = [  
5     fruit  
6     for fruit in fruits  
7     if (  
8         fruit.count("a")  
9         + fruit.count("e")  
10        + fruit.count("i")  
11        + fruit.count("o")  
12        + fruit.count("u")  
13    )  
14    == 2  
15 ]  
16  
17 print(fruits_with_only_two_vowels)
```

q214.py

**Question 16**

Faire une liste qui contient chaque fruit avec plus de 5 caractères

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4 print([fruit for fruit in fruits if (len(fruit) > 5)])
```

q215.py

**Question 17**

Faire une liste qui contient chaque fruit avec exactement 5 caractères

Code python :





```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 print([fruit for fruit in fruits if (len(fruit) == 5)])
```

q216.py

### Question 18

Faire une liste qui contient des fruits qui ont moins de 5 caractères

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 print([fruit for fruit in fruits if (len(fruit) < 5)])
```

q217.py

### Question 19

Faites une liste contenant le nombre de caractères de chaque fruit. Les résultats seraient 5, 4, 10, etc...

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 print([len(fruit) for fruit in fruits])
```

q218.py

### Question 20

Créez une variable nommée fruits\_avec\_lettre\_a qui contient une liste des seuls fruits contenant la lettre "a"

Code python :



```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 fruits_with_letter_a = [fruit for fruit in fruits if "a" in fruit]  
5 print(fruits_with_letter_a)
```

q219.py

### Question 21

Créer une variable nommée `even_numbers` qui ne contiendra que les nombres pairs

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 even_numbers = [number for number in numbers if number % 2 == 0]  
5 print(even_numbers)
```

q220.py

### Question 22

Créer une variable nommée `nombres_impairs` qui ne contient que les nombres impairs

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 odd_numbers = [number for number in numbers if number % 2 != 0]  
5 print(odd_numbers)
```

q221.py

### Question 23

Créer une variable nommée `nombres_positifs` qui ne contient que les nombres positifs

Code python :



```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 positive_numbers = [number for number in numbers if number > 0]  
5 print(positive_numbers)
```

q222.py

### Question 24

Créer une variable nommée `nombre_négatifs` qui ne contient que les nombres négatifs

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 negative_numbers = [number for number in numbers if number < 0]  
5 print(negative_numbers)
```

q223.py

### Question 25

Utiliser une compréhension de liste avec un conditionnel afin de produire une liste de nombres avec 2 chiffres ou plus

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 print([number for number in numbers if number > 9 or number < -9])
```

q224.py

### Question 26

Créez une variable nommée `nombre_carré` qui contient la liste des nombres avec chaque élément au carré. La sortie est `[4, 9, 16, etc...]`



Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 numbers_squared = [number ** 2 for number in numbers]  
5 print(numbers_squared)
```

q225.py

### Question 27

Créez une variable nommée `nombre_impairs_négatifs` qui ne contient que les nombres qui sont à la fois impairs et négatifs.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 odd_negative_numbers = [number for number in numbers if number < 0 and  
    ↪ number % 2 != 0]  
5 print(odd_negative_numbers)
```

q226.py

### Question 28

Créez une variable nommée `nombre_plus_5`. Dans cette variable, renvoyez une liste contenant chaque nombre plus cinq.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
    ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
    ↪ -2, 5, -9]  
3  
4 numbers_plus_5 = [number + 5 for number in numbers]  
5 print(numbers_plus_5)
```

q227.py

### Question 29

Créez une variable nommée `primes` qui est une liste contenant les nombres premiers de la liste des nombres. \*Astuce : vous pouvez créer ou trouver une fonction d'aide



qui détermine si un nombre donné est premier ou non.

Code python :

```
1 fruits = ["mango", "kiwi", "strawberry", "guava", "pineapple",  
  ↪ "mandarin orange"]  
2 numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 19, 23, 256, -8, -4,  
  ↪ -2, 5, -9]  
3  
4  
5 def is_prime(num):  
6     prime_check = False  
7     if num > 1:  
8         for i in range(2, num):  
9             if (num % i) == 0:  
10                prime_check = False  
11                break  
12        else:  
13            prime_check = True  
14    return prime_check  
15  
16  
17 primes = [number for number in numbers if is_prime(number)]  
18 print(primes)
```

q228.py