

<b>INFORMATIQUE - EXERCICES D'ENTRAÎNEMENT 1 - CORRIGÉS</b>
---

Corrigés des :

1 - Exos sur les boucles for ... page 2

2 - Exos sur les constructions de listes avec l'instruction for ... page 4

3 - Exos sur les instructions conditionnelles if, elif, else ... page 5

4 - Exos sur les boucles while ... page 7

5 - Exos sur les chaînes de caractères ... page 10

6 - Exos sur les fonctions ... page 14

## 1 - BOUCLES FOR

---

```
from math import *

#####
# EXERCICE 1

for machin in range(50): # le compteur ne doit pas forcément s'appeler 'k'
    print('Je dois ranger mon bureau')

#####
# EXERCICE 2

Somme = 0
for entier in range(4,2024):
    Somme +=entier # synonyme de Somme = Somme + entier
print(Somme)

#####
# EXERCICE 3

Somme = 0
for entier in range(4,2024):
    Somme +=entier**2
print(Somme)

#####
# EXERCICE 4

Somme = 0
for entier in range(5,402,3):
    Somme +=entier
print(Somme)

#####
# EXERCICE 5

print('Choisissez un entier')
N =int(input())
S = 0
for entier in range(1,N+1):
    S +=1/entier
print(S)
```

```
#####
```

```
# EXERCICE 6
```

```
# QUESTION 1
```

```
print(' Choisissez un entier ')\nN =int(input())\nS =0\nfor entier in range(1,N+1):\n    S +=1/(entier**2)\nprint(S)
```

```
# QUESTION 2
```

```
print(' Choisissez un entier ')\nN =int(input())\nS =0\nfor entier in range(1,N+1):\n    S +=1/(entier**2)\nprint(((pi**2)/6)-S)
```

```
#####
```

```
# EXERCICE 7
```

```
from random import randint
```

```
print(' Choisissez un nombre de lancers ')\nNblancers = int(input())\n\nfor compteur in range(Nblancers):\n    print(randint(1,6))
```

```
#####
```

```
# EXERCICE 8
```

```
from random import randint
```

```
print(' Choisissez un nombre de tirages ')\nNbtirages = int(input())\n\nalphabet = 'abcdefghijklmnopqrstuvwxyz'\n\nfor compteur in range(Nblancers):\n    rang = randint(0,25)\n    print(alphabet[rang])
```

```
#####
# EXERCICE 9

animaux = ['chien', 'mouton', 'chat', 'girafon', 'lapin', 'pingouin', 'pomme-purée']

print('Choisissez un entier entre 1 et 7')
rang = int(input())

print('Le mot ', animaux[rang-1], ' possède ', len(animaux[rang-1]), ' caractères .')
```

---

## 2 - CONSTRUCTIONS DE LISTES AVEC L'INSTRUCTION FOR

---

```
#####
# EXERCICE 10

Lexo10 = [3*n for n in range(1,34)]

#####
# EXERCICE 11

from random import randint

Lexo11 = [randint(1,6) for oiseau in range(20)]

#####
# EXERCICE 12

Lexo12 = [entier % 2 for entier in range(8)]

#####
# EXERCICE 13

Lexo13 = [entier // 2 for entier in range(8)]

#####
# EXERCICE 14

# QUESTION 1

Lexo14 = [0.1 for bidule in range(10)]

# QUESTION 2: ok, 10 * 0.1 = 1...

# QUESTION 3

Somme = 0
for elem in Lexo14:
    Somme += elem
print(Somme) # la valeur affichée n'est pas exactement 1...

# QUESTION 4: la bonne réponse est la réponse D!
```

### 3 - INSTRUCTIONS CONDITIONNELLES IF, ELIF, ELSE

```
#####  
# EXERCICE 15
```

```
print(' Choisissez un premier nombre')  
x = float(input())
```

```
print(' Choisissez un second nombre')  
y = float(input())
```

```
if x <=y:  
    print(x)  
else:  
    print(y)
```

```
#####  
# EXERCICE 16
```

```
print(' Choisissez un premier entier pair')  
a = int(input())
```

```
print(' Choisissez un second entier pair')  
b = int(input())
```

```
if (a%2 !=0) or (b%2 !=0):  
    print('ERREUR!')  
else:  
    print('La moyenne de ',a,' et ',b,' est : ',(a+b)/2)
```

```
#####  
# EXERCICE 17
```

```
for k in range(1,21):  
    N =11*k  
    if N % 7 ==0:  
        print(N)
```

```
#####  
# EXERCICE 18
```

```
for k in range(1,21):  
    N =7*k  
    if N % 3 ==0:  
        print(N,' *')  
    else:  
        print(N)
```

```
#####
# EXERCICE 19
```

```
print(' Choisissez une chaîne de caractères ')
str1 = input()

print(' Choisissez une autre chaîne de caractères ')
str2 = input()
```

```
if len(str1) < len(str2):
    print('Le mot le plus court est: ', str1)
elif len(str2) < len(str1):
    print('Le mot le plus court est: ', str2)
else:
    print('match nul')
```

```
#####
# EXERCICE 20
```

```
print(' Choisissez un entier ')
entier = int(input())

if entier % 2 == 0:
    print('Ce nombre est pair')
elif entier % 3 == 0:
    print('Ce nombre est impair mais multiple de 3')
else:
    print("Ce nombre n'est ni pair ni multiple de 3")
```

```
#####
# EXERCICE 21
```

```
Lsolutions = [ ] # initialisation de la liste des solutions
```

```
SommeTotale = 0 # calcul de la somme des entiers de 1 à 26
```

```
for k in range(1,27):
    SommeTotale += k
```

```
for a in range(1,27):
    for b in range(1,27):
        if a != b:
            produit = a*b
            if produit == SommeTotale - a - b:
                Lsolutions += [(a,b)]
print(Lsolutions)
```

```
# Les couples (a,b) solutions sont (15, 21)... et (21, 15)
```

## 4 - BOUCLES WHILE

```
from math import *

#####
# EXERCICE 22

for compteur in range(2,101,2):
    print(compteur)

entier = 2
while entier <=100:
    print(entier)
    entier +=2

#####
# EXERCICE 23

print(' Choisissez un entier ')
n = int(input())

while n > 0:
    print("C'est dans ",n,"ans je m'en irai j'entends le loup le renard chanter")
    n -=1

# En corrigeant la faute d'orthographe de la dernière ligne

print(' Choisissez un entier ')
n = int(input())

while n > 1:
    print("C'est dans ",n,"ans je m'en irai j'entends le loup le renard chanter")
    n -=1

print("C'est dans 1 an je m'en irai j'entends le loup le renard chanter")

#####
# EXERCICE 24

print(' Choisissez un nombre positif ')
nbre = float(input())

entier = 0

while entier*entier <=nbre:
    entier +=1

print(entier - 1)
```

```
#####  
# EXERCICE 25
```

```
print(' Choisissez un nombre')  
X = float(input())
```

```
Somme =1  
rang =1
```

```
while Somme <=X:  
    rang +=1  
    Somme +=1/rang
```

```
print(rang)
```

```
#####  
# EXERCICE 26
```

```
print(' Choisissez un nombre (epsilon)')  
epsi = float(input())
```

```
Somme =1  
rang =1
```

```
while ((pi**2)/6)-Somme >=epsi:  
    rang +=1  
    Somme +=1/rang**2
```

```
print(rang,Somme)
```

```
#####  
# EXERCICE 27
```

```
print(' Choisissez un entier')  
entier = int(input())
```

```
div =2
```

```
while (entier % div !=0) and (div < entier):  
    div +=1
```

```
print('Le plus petit diviseur (>1) de ',entier,' est ',div)
```



```
#####  
# EXERCICE 28
```

```
from random import random
```

```
print(' Choisissez un entier ')  
N =int(input())
```

```
ListPts =[]
```

```
while len( ListPts) < N:  
    x =random()  
    y =random()  
    if x**2 +y**2 <=1:  
        ListPts +=[(x,y)]
```

```
print( ListPts)
```

```
#####  
# EXERCICE 29
```

```
from random import randint
```

```
print(' Choisissez un entier au plus égal à 20')  
N =int(input())
```

```
if N >20:
```

```
    print('ERREUR')
```

```
else:
```

```
    ListNbre =[]
```

```
    while len(ListNbre) < N:
```

```
        nalea = randint(1,1000)
```

```
        naleacopie =nalea
```

```
        c = nalea // 100 # c = chiffre des centaines de nalea
```

```
        nalea =nalea - c*100
```

```
        d = nalea // 10 # d = chiffre des dizaines de nalea
```

```
        nalea =nalea - d*10
```

```
        u = nalea # u = chiffre des unités de nalea
```

```
        if c+d+u ==6:
```

```
            ListNbre +=[naleacopie]
```

```
print( ListNbre)
```

## 5 - CHAÎNES DE CARACTÈRES

```
#####
# EXERCICE 30
```

```
print('Tape un mot')
MOT =input()
print("L' initiale de ton mot est ",MOT[0])
```

```
#####
# EXERCICE 31
```

```
print('Tape un mot de 4 lettres ou plus')
MOT =input()
if len(MOT) <4:
    print("Et alors ! Tu sais pas compter jusqu'à 4!")
else:
    print('Les 4 premières lettres de ton mot sont ',MOT[:4])
```

```
#####
# EXERCICE 32
```

```
print('Tape un mot')
MOT =input()
NbE =0
for lettre in MOT:
    if lettre == 'e':
        NbE +=1
print('Ton mot contient ',NbE,' fois la lettre e')
```

```
#####
# EXERCICE 33
```

```
print('Ecris quelque chose')
phrase =input()
phrase2 = ''

for k in range(len(phrase)):
    if phrase[k] not in 'aeiouyAEIOUYéeàêâû':
        phrase2 +=phrase[k]
    else:
        phrase2 += 'o'

print(phrase2)
```

```
#####
# EXERCICE 34
```

```
Lval = []
for k in range(1,11):
    Lval +=[str(k)]
Lval +=['V', 'D', 'R']
```

```
#####
# EXERCICE 35
```

```
Lcoul = ['Pique', 'Coeur', 'Carreau', 'Trefle'] # ok, pas trop dur!
```

```
#####
# EXERCICE 36
```

```
Ljeu = []
for couleur in Lcoul:
    for valeur in Lval:
        Ljeu += [valeur + ' de ' + couleur]
```

```
#####
# EXERCICE 37
```

```
from random import randint
Lcartes = []
while len(Lcartes) < 5:
    nalea = randint(1,52)
    if nalea not in Lcartes:
        Lcartes += [nalea]
```

```
Lcartes2 = [Ljeu[elem] for elem in Lcartes]
print(Lcartes2)
```

```
#####
# EXERCICE 38
```

```
# Programme de l'exo 37 + les lignes de code suivantes
```

```
Test = True
pos = 0
while pos < 5 and Test:
    if Lcartes2[pos][-3] == Lcartes2[0][-3]: # on compare les 3 derniers caractères
        pos += 1
    else:
        Test = False
if Test:
    print('Couleur')
```

```
#####
# EXERCICE 39
```

```
print('Ecris quelque chose')
phrase = input()

print(phrase[0::2]) # ne conserve qu'une lettre sur deux en partant de la 1ère
```

```
#####  
# EXERCICE 40  
  
# CORRIGE — VERSION 1  
  
from random import *  
  
animaux = ['chien', 'mouton', 'chat', 'girafon', 'lapin', 'pingouin', 'presse-purée']  
  
compar = ["a-t-il plus de pattes", "est-il plus grand", "vole-t-il plus vite",  
"mange-t-il plus de carottes", "est-il aussi affectueux"]  
  
for compteur in range(3):  
    anim1 = randint(0,6)  
    anim2 = randint(0,6)  
    while anim2 == anim1:  
        anim2 = randint(0,6)  
    compar1 = randint(0,4)  
  
    question = 'Le ' + animaux[anim1] + ' ' + compar[compar1] + ' que le ' + animaux[anim2] + ' ?'  
  
    print(question)
```

*# CORRIGE — VERSION 2 — EN EXCLUSIVITE MONDIALE, LA VERSION QUI PARLE!!!!*

*# Pour "apprendre à parler" à votre machine, taper au préalable dans le SHELL:*

*# >>> pip install gtts*

*# Ceci fait, utiliser les instructions suivantes :*

*# from gtts import gTTS*

*# import os*

*# tts = gTTS(text="Bonjour! Je suis une machine qui parle.", lang='fr')*

*# tts.save("question.mp3")*

*# os.system("start question.mp3")*

*# Mise en application, avec la version parlante*

**from random import \***

**from gtts import gTTS**

**import os**

**animaux = ['chien', 'mouton', 'chat', 'girafon', 'lapin', 'pingouin', 'presse-purée']**

**compar = ["a-t-il plus de pattes", "est-il plus grand", "vole-t-il plus vite",  
"mange-t-il plus de carottes", "est-il aussi affectueux"]**

**anim1 = randint(0,6)**

**anim2 = randint(0,6)**

**while anim2 == anim1:**

**anim2 = randint(0,6)**

**compar1 = randint(0,4)**

**QUESTION = 'Le ' + animaux[anim1] + ' ' + compar[compar1] + ' que le ' + animaux[anim2] + ' ?'**

**tts = gTTS(text = QUESTION, lang = 'fr')** *# remplacer 'fr' par 'da' pour un superbe accent danois, ou  
par 'nl' pour un accent néerlandais inoubliable*

**tts.save("question.mp3")**

**os.system("start question.mp3")**

## 6 - FONCTIONS

```
#####  
# EXERCICE 41
```

```
def fonc1(x):  
    return x**2+x+1
```

```
#####  
# EXERCICE 42
```

```
# Corrigé 1, version "naturelle "
```

```
def NB4(L):  
    compte_les_4 =0  
    for elem in L:  
        if elem ==4:  
            compte_les_4 +=1  
    return compte_les_4
```

```
# Corrigé 2, version "compacte"
```

```
def NB4bis(L):  
    return len([elem for elem in L if elem ==4])
```

```
#####  
# EXERCICE 43
```

```
# Corrigé 1, version "naturelle "
```

```
def NB(L,n):  
    compte_les_n =0  
    for elem in L:  
        if elem ==n:  
            compte_les_n +=1  
    return compte_les_n
```

```
# Corrigé 2, version "compacte"
```

```
def NBbis(L,n):  
    return len([elem for elem in L if elem ==n])
```

```
#####  
# EXERCICE 44
```

```
def Pairs(L):  
    nbpairs = 0  
    listpairs = []  
    for k in range(len(L)):  
        if L[k] %2 ==0:  
            nbpairs +=1  
            listpairs +=[k]  
    return nbpairs, listpairs
```

```
#####  
# EXERCICE 45
```

```
from math import *
```

```
# Corrigé 1, version "naturelle"
```

```
def Vapproch(x,y, epsilon):  
    if abs(y-x) <=epsilon:  
        return True  
    else:  
        return False
```

```
# Corrigé 2, version "compacte"
```

```
def Vapprochbis(x,y, epsilon):  
    return abs(y-x) <=epsilon
```

```
#####  
# EXERCICE 46
```

```
def MinMax(L):  
    mini, maxi =L[0], L[0]  
    for elem in L:  
        if elem < mini:  
            mini =elem  
        elif elem > maxi:  
            maxi =elem  
    return mini, maxi
```

```
#####
```

```
# EXERCICE 47
```

```
# QUESTION 1
```

```
def Fexo47(N):  
    u = 2  
    for n in range(N):  
        u = (2*u+1)/3  
    return u
```

```
# QUESTION 2
```

```
print([Fexo47(rang) for rang in range(20)])
```

```
#####
```

```
# EXERCICE 48
```

```
# QUESTION 1
```

```
def Fexo48(N):  
    u, v = 1, 3  
    if N == 0:  
        return u  
    elif N == 1:  
        return v  
    else:  
        for k in range(N-1):  
            u, v = v, v - u + 1  
    return v
```

```
# QUESTION 2
```

```
print([Fexo48(rang) for rang in range(20)])
```

```
#####
```

```
# EXERCICE 49
```

```
# QUESTION 1
```

```
def Fexo49(N):  
    u, v = 2, 10  
    for n in range(N):  
        u, v = (2*u+v)/3, (u+3*v)/4  
    return u,v
```

```
# QUESTION 2
```

```
print([Fexo49(rang) for rang in range(20)])
```



```
#####
```

```
# EXERCICE 50
```

```
from math import *
```

```
# QUESTION 1
```

```
def Dper(a,b,c):
    return (a+b+c)/2
```

```
# QUESTION 2
```

```
def AireT(a,b,c):
    p = Dper(a,b,c)
    return sqrt(p*(p-a)*(p-b)*(p-c))
```

```
#####
```

```
# EXERCICE 51
```

```
# Corrigé 1, version "naturelle"
```

```
def compteVoy(chaine):
    nbVoy = 0
    for lettre in chaine:
        if lettre in 'aeiouy':
            nbVoy += 1
    return nbVoy
```

```
# Corrigé 2, version "compacte"
```

```
def compteVoybis(chaine):
    return len([elem for elem in chaine if elem in 'aeiouy'])
```

```
#####
```

```
# EXERCICE 52
```

```
def Only_Voy(chaine):
    mon_booleen_a_moi = True
    for lettre in chaine:
        if lettre not in 'aeiouy':
            mon_booleen_a_moi = False
    return mon_booleen_a_moi
```

```
#####
```

```
# EXERCICE 53
```

```
# Corrigé 1, version "naturelle"
```

```
def Egalite_de_Voy(chaine1, chaine2):
    if compteVoy(chaine1) == compteVoy(chaine2):
        return True
    else:
        return False
```

```
# Corrigé 2, version "compacte"
```

```
def Egalite_de_Voybis(chaine1, chaine2):
```

```
#####
```

```
# EXERCICE 54
```

```
def Afrikaans(chaine):
    test = True
    for carac in chaine:
        if carac in 'CQXZcqxz':
            test = False
    return test
```

```
#####
```

```
# EXERCICE 55
```

```
def codage(TXT):
    return TXT[0::2], TXT[1::2]

def decodage(TXT1, TXT2):
    res = ''
    for k in range(len(TXT1)-1):
        res += TXT1[k] + TXT2[k]
    if len(TXT1) == len(TXT2):
        res += TXT1[len(TXT1)-1] + TXT2[len(TXT1)-1]
    else:
        res += TXT1[len(TXT1)-1]
    return res
```

```
#####
```

```
# EXERCICE 56
```

```
def Deci(n):
    if n == 0:
        return [0]
    else:
        Ldeci = []
        while (n > 0):
            chif = n % 10
            Ldeci = [int(chif)] + Ldeci
            n = (n - chif) / 10
        return Ldeci
```

```
#####
```

```
# EXERCICE 57
```

```
def Bin(n):
    if n == 0:
        return 0
    else:
        Lbin = []
        while (n > 0):
            chif = n % 2
            Lbin = [str(int(chif))] + Lbin
            n = (n - chif) / 2
        EcriBin = ''
        for elem in Lbin:
            EcriBin += elem
```

```
#####
# EXERCICE 58
```

```
def BinDeci(n):
    EcriBin = str(n)
    N = len(EcriBin)
    Nbdec = 0
    for k in range(N):
        Nbdec += int(EcriBin[k]) * 2**(N-1-k)
    return Nbdec
```

```
#####
# EXERCICE 59
```

```
from math import *
```

```
# QUESTION 1
```

```
def Tprem(n):
    if n < 2:
        Test = False
    else:
        Test = True
        div = 2
        while Test and div <= sqrt(n):
            if n % div == 0:
                Test = False
            else:
                div += 1
    return Test
```

```
# QUESTION 2
```

```
# Corrigé 1, version "naturelle"
```

```
def Listeprem(n):
    L = []
    for entier in range(2, n+1):
        if Tprem(entier):
            L += [entier]
    return L
```

```
# Corrigé 2, version "compacte"
```

```
def Listeprembis(n):
    return [nombre for nombre in range(2, n+1) if Tprem(nombre)]
```

```
#####
```

```
# EXERCICE 60
```

```
# QUESTION 1
```

```
def Fermat(n):
    return 2**(2**n)+1
```

```
# QUESTION 2
```

```
F5 =Fermat(5)
div =2
while F5 % div !=0:
    div +=1
print(div, ' est un diviseur du nombre de Fermat F5')
```

```
#####
```

```
# EXERCICE 61
```

```
# QUESTION 1
```

```
def Mersenne(n):
    return 2**n-1
```

```
# QUESTION 2
```

```
Lstprem_2_1000 =Listeprembis(1000)
```

```
# QUESTION 3
```

```
rang =0
while Tprem(Mersenne(Lstprem_2_1000[rang])):
    rang +=1
print(rang)
# Le nombre recherché est Mersenne(11). Il est égal à 2047 = 23 * 89
#####
# EXERCICE 62
```

```
# QUESTION 1
```

```
def Somcubchif(n):
    listchiff =[]
    for compteur in range(3):
        chif =n%10
        n =(n - chif)/10
        listchiff =[int(chif)] + listchiff
    Somcub =0
    for elem in listchiff :
        Somcub +=elem**3
    return Somcub
```

```
# QUESTION 2
```

```
# L'instruction ci-dessous construit la liste des nombres d'Armstrong strictement inférieurs à 1000
ListArmstrong =[entier for entier in range(1,1000) if entier ==Somcubchif(entier)]
```

```
#####
```

```
# EXERCICE 63
```

```
# QUESTION 1
```

```
def Divpro(n):
    return [divi for divi in range(1, int(n/2)+1) if n % divi == 0]
```

```
# QUESTION 2
```

```
def Ami(a,b):
    DPA, DPB = Divpro(a), Divpro(b)
    SomDivA, SomDivB = 0, 0
    for elem in DPA:
        SomDivA += elem
    for elem in DPB:
        SomDivB += elem
    return SomDivA == b and SomDivB == a
```

```
#####
```

```
# EXERCICE 64
```

```
# QUESTION 1
```

```
# Utilise la fonction Tprem de l'exo 59, qui sert à déterminer
# si un entier est premier
```

```
def Tcarmi(N):
    BOOL = True
    if Tprem(N):
        BOOL = False
    if BOOL:
        k=2
        reste = 0
        while BOOL and k<N:
            reste = (-k+k**N) % N
            if reste != 0:
                BOOL = False
            k = k + 1
    return BOOL
```

```
# QUESTION 2
```

```
def Cherchecarmi(a,b):
    return [nb for nb in range(a,b) if Tcarmi(nb)]
```

```
# 561 unique nb de Carmi < 1000; puis il y en a 6 qui sont <10 000
# (à savoir 1105, 1729, 2465, 2821, 6601 et 8911)
```

```
#####
```

```
# EXERCICE 65
```

```
def Racines(N):
    return [cos(2*k*pi/N) for k in range(N)], [sin(2*k*pi/N) for k in range(N)]
```

```
#####
# EXERCICE 66
```

```
from math import *
from time import *

def Listcos(p):
    t0 =time() # On lance le chronomètre
    u =1
    L =[ ]
    for k in range(p):
        test =True
        while test :
            if abs(cos(u)+1) <10**(-k):
                L =L +[u]
                test =False
            u =u +1
    return L, time()-t0
```

```
# time()-t0 = durée d'exécution du programme
```

```
#####
# EXERCICE 67
```

```
# QUESTION 1
```

```
def fiboterm(p):
    u, v =0, 1
    for k in range(p):
        u, v =v, u +v
    return u

def fibonacci(p):
    return [fiboterm(k) for k in range(p+1)]
```

```
# QUESTION 2
```

```
def recherche(n):
    if n ==0:
        return 0
    elif n ==1:
        return 2
    else:
        k =2
        while fibonacci(k) <=n:
            k +=1
        k -=1
    return k
```

```
#####  
# EXERCICE 68
```

```
# QUESTION 1
```

```
def F68(N):  
    Som = 0  
    for k in range(N+1):  
        Som += ((-1)**k)/(2*k+1)  
    return Som
```

```
# QUESTION 2
```

```
print(' Choisis un entier N')  
N = int(input())  
print ([F68(compteur) for compteur in range(N+1)])
```

```
#####  
# EXERCICE 69
```

```
# Les questions 1 et 2 sont celles de l'exo 59
```

```
def Valpadic(n,p):  
    k=0  
    while n%(p**(k+1)) !=0:  
        k +=1  
    return k
```

```
#####  
# EXERCICE 72
```

```
# QUESTION 2
```

```
def creerReseauVide(n):  
    return [n, []]
```

```
# QUESTION 3
```

```
def estUnLienEntre(paire, i, j):  
    return (paire == [i, j]) or (paire == [j, i])
```

```
# QUESTION 4
```

```
def sontAmis(reseau, i, j):  
    for paire in reseau[1]:  
        if estUnLienEntre(paire, i, j):  
            return True  
    return False
```

```
# QUESTION 5
```

```
def declareAmis(reseau, i, j):  
    if not sontAmis(reseau, i, j):  
        reseau[1].append([i, j])
```

*# QUESTION 6*

```
def listeDesAmisDe(reseau, i):  
    amis = []  
    for paire in reseau[1]:  
        if paire[0] == i:  
            amis.append(paire[1])  
        elif paire[1] == i:  
            amis.append(paire[0])  
    return amis
```



```

#####
# EXERCICE 70

# Le scrabble, dans sa version Galloise, c'est plus fun

Score=0

Test =0
while (Test ==0):
    print(' Saisissez un mot gallois (pas de K, de Q, de V, de X ni de Z) en majuscules!')
    mot =input()
    Test =1
    for k in range(len(mot)):
        if mot[k] not in 'ABCDEFGHIIJLMNOPRSTUWY':
            Test =0
    if (Test ==0):
        print('Try again, loser!')

k =0
while (k < len(mot)):
    if (mot[k] =='A') | (mot[k] =='E') | (mot[k] =='I') | (mot[k] =='Y') | (mot[k] ==
'O') | (mot[k] =='W'):
        Score =Score +1
        k =k +1
    elif (mot[k] =='G') | (mot[k] =='U'):
        Score =Score +2
        k =k +1
    elif (mot[k] =='S') | (mot[k] =='B') | (mot[k] =='M'):
        Score =Score +3
        k =k +1
    elif (mot[k] =='H'):
        Score =Score +4
        k =k +1
    elif (mot[k] =='P'):
        Score =Score +5
        k =k +1
    elif (mot[k] =='J'):
        Score =Score +8
        k =k +1
    elif (mot[k] =='D'):
        if (k ==len(mot)-1):
            Score =Score +1
            k =k +1
        elif (mot[k+1] =='D'):
            Score =Score +1
            k =k +2
        else:
            Score =Score +1
            k =k +1
    elif (mot[k] =='C'):
        if (k ==len(mot)-1):
            Score =Score +4
            k =k +1
        elif (mot[k+1] =='H'):
            Score =Score +5

```

```
    else:
        Score = Score + 4
        k = k + 1
elif (mot[k] == 'F'):
    if (k == len(mot)-1):
        Score = Score + 2
        k = k + 1
    elif (mot[k+1] == 'F'):
        Score = Score + 4
        k = k + 2
    else:
        Score = Score + 2
        k = k + 1
elif (mot[k] == 'T'):
    if (k == len(mot)-1):
        Score = Score + 3
        k = k + 1
    elif (mot[k+1] == 'H'):
        Score = Score + 4
        k = k + 2
    else:
        Score = Score + 3
        k = k + 1
elif (mot[k] == 'L'):
    if (k == len(mot)-1):
        Score = Score + 2
        k = k + 1
    elif (mot[k+1] == 'L'):
        Score = Score + 5
        k = k + 2
    else:
        Score = Score + 2
        k = k + 1
elif (mot[k] == 'N'):
    if (k == len(mot)-1):
        Score = Score + 1
        k = k + 1
    elif (mot[k+1] == 'G'):
        Score = Score + 10
        k = k + 2
    else:
        Score = Score + 1
        k = k + 1

if (k == len(mot)-1):
    Score = Score + 1
    k = k + 1
elif (mot[k+1] == 'G'):
    Score = Score + 10
    k = k + 2
else:
    Score = Score + 1
    k = k + 1
```

```
elif (mot[k] == 'R'):  
    if (k == len(mot)-1):  
        Score = Score + 1  
        k = k + 1  
    elif (mot[k+1] == 'H'):  
        Score = Score + 10  
        k = k + 2  
    else:  
        Score = Score + 1  
        k = k + 1  
  
print('Le mot ', mot, ' rapporte ', Score, ' points au Scrabble gallois')
```