

Exercice 1 : jouer au Mastermind

1) Version Allégée

Voici le code, avec le mot LAVE stocké en mémoire.

```
mot='lave'
print('\nPour interrompre le programme, tapez STOP')

corr=0          # Nbre de lettres bien placées
m=mot.upper()  # On passe le mot en majuscules

# Boucle Principale
while corr <4:
    entree=''

    # on s'assure que l'entrée a la bonne longueur
    while len(entree)!=4:
        entree=input('Mot de 4 lettres : ')

    e=entree.upper() # On passe entree en majuscules
    # on teste la condition de sortie
    if e=='STOP':
        break

    # on compte les lettres bien placées
    corr=0
    for i in range(4):
        if e[i]==m[i]:
            corr = corr + 1
    print(corr,'lettre(s) bien placée(s)')

    if corr==4:
        print("C'est le bon mot, bravo !")
```

Avec les entrées LOUP, PALE et LEVA, on a bien les résultats attendus.

2) Version Kouign Amann

Voici le programme modifié.

```
mot='chat'
print('Pour interrompre le programme, tapez STOP')

corr=0          # Nbre de lettres bien placées
depl=0          # Nbre de lettres mal placées
m=mot.upper()  # On passe le mot en majuscules

# Boucle Principale
while corr <4:
    l_m=[e for e in m] # Liste des lettres du mot
    entree=''

    # on s'assure que l'entree a la bonne longueur
    while len(entree)!=4:
        entree=input('Mot de 4 lettres : ')

    e=entree.upper() # On passe entree en majuscules
    l_e=[c for c in e] # Liste de ses lettres

    # on teste la condition de sortie
    if e=='STOP':
        break

    # on compte les lettres bien placées
    corr=0
    for i in range(4):
        if e[i]==m[i]:
            corr = corr + 1
            l_e.remove(e[i])
            l_m.remove(m[i])
    print(corr,'lettre(s) bien placée(s)')

    if corr==4:
        print("C'est le bon mot, bravo !")

    # on compte les lettres mal placées
    depl=0
    for i in l_e:
        if i in l_m:
            depl = depl + 1
            l_m.remove(i)
    if depl!=0:
        print(depl,'lettre(s) mal placée(s)')
```

Avec les entrées TATA et SHAH, on a les résultats attendus.

Exercice 2 : écriture d'un entier en base b

Partie A : préliminaires

- 1) On s'aperçoit que la commande `str()` convertit les données en chaînes de caractères. Ainsi, les deux dernières instructions permettent de créer les chaînes "421" et "214".
- 2) On tape : `c=str(a)+b`.

Partie B : Un exemple pour se faire la main

- 1) Voici le tableau complété (il est encore plus beau) :

	n	q	r	resultat
Initialisation	13			' '
Étape 1	13	6	1	
	6			'1'
Étape 2	6	3	0	
	3			'01'
Étape 3	3	1	1	
	1			'101'
Étape 4	1	0	1	
	0			'1101'

Notons que l'on s'arrête lorsque $n=0$ (c'est le dernier quotient).

- 2) (a) $q, r = n//2, n\%2$

(b) `resultat = str(r)+resultat`

- 3) On s'arrête lorsque $n=0$.

Ainsi, on écrira une boucle commençant par `while n>0:`

- 4)


```
# On initialise nos variables
n = 13
resultat = ''

# La boucle pour effectuer la décomposition
while n>0:
    q,r = n//2,n%2
    resultat = str(r)+resultat
    n = q

# L'affichage final
print("L'entier 13 s'écrit",resultat,"en base 2")
```

Partie C : Généralisation du programme

Voici un premier jet fonctionnant pour $b < 11$.

```
# On initialise nos variables
n=int(input("Entrez un l'entier à coder : "))
b=int(input("Entrez la base : "))

N = n
resultat = ''

# La boucle pour effectuer la décomposition
while n>0:
    q,r = n//b, n%b
    resultat = str(r)+resultat
    n = q

# L'affichage final
print("\nL'entier",N,"s'écrit",resultat,"en base",b)
```

Maintenant, une belle astuce pour gérer les bases entre 11 et 36 : on utilise une chaîne dont les caractères correspondent aux 36 chiffres en base 36, leurs indices dans la chaîne correspondant à leur valeur en base 10.

```
# On initialise nos variables
code='0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ'
n=int(input("Entrez un l'entier à coder : "))
b=int(input("Entrez la base : "))

N = n
resultat = ''

# La boucle pour effectuer la décomposition
while n>0:
    q,r = n//b,n%b
    resultat = code[r]+resultat
    n = q

# L'affichage final
print("\nL'entier",N,"s'écrit",resultat,"en base",b)
```

C'est très joli, ça fonctionne bien et surtout cela permet de briller en société.