

THOMAS

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

print("Tu veux un lysopaine ?")
reponse = raw_input()
reponse = str(reponse.lower())

if reponse == "oui" :
    import re

    f= open("poeme.txt", "r")

    poeme = f.read()

    poeme = poeme.lower()

    #transforme les a en aaaah
    lyso = re.sub("a", "aaaah", poeme)

    # boucle qui imprime 1 caractère sur 2 en capitale
    i = 1
    poeme = ""
    while i < len(lyso):
        poeme = poeme + lyso[i-1].upper() + lyso[i].lower()
        i += 2

    print(poeme)
    f.close()

else:
    print("Ben tant pis pour toi.")
```

Je voudrais créer un programme qui transforme les "a" en "aaaah",et qui transforme une lettre sur 2 en Capitale, pour donner un style ado en train de muer.

GESINE

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

with open('poetry','r') as lecture:
    poyme = lecture.read()

# transforme les 'a' en 'aaaah'
poyme = poyme.replace("a","aaaah")

# transforme les 'A' en 'AAAHH'
poyme = poyme.replace("A","AAAHH")

print(poyme)
```

REPLACEMENT

REPLACEMENT

REPLACEMENT

Remplacer tout les 'o' par 0 et tout les 'l' par 1

REPLACEMENT

REPLACEMENT

```
#!/usr/bin/env python
#coding: utf-8

print("Quel est ton nom, poète.esse ?")
nom = raw_input()

import re

f= open("poeme.txt", "r+")

poeme = f.read()

#poeme = re.sub('a', "aouuuu", poeme)
#poeme = re.sub('[aA]', r"\louuuu", poeme)
#poeme = re.sub('[gG]', r"\lrrrrrrrrrr", poeme)

nom = re.sub('[aA]', r"\louuuu", nom)
nom = re.sub('[gG]', r"\lrrrrrrrrrr", nom)
print(poeme)
print("Signé, " + nom + " le loup grrrrrrrrraouuuuou")

f.close()
```

Un programme qui - demande le nom de l'utilisateur qui sera enregistré comme nom de poète - rajoute aouuuu quand il y a a lettre a - rajoute grrrrrrrrrr quand il y a la lettre g

REPLACEMENT

REPLACEMENT

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

import re

f= open("poeme.txt", "r+")
poeme = f.read()

if plue == "oui":
    ouï = re.sub('a|e|i|o|u', "i", poeme, flags=re.IGNORECASE)
    print(ouï)
    print("Quel temps pourri...")
else:
    print("Chouette alors !")
f.close()
```

Demander est ce qu'il pleut? oui: changer tous les a e i o u < > | non: print chance

REPLACEMENT

REPLACEMENT

REPLACEMENT

REPLACEMENT

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

with open('poetry', 'r') as teteLecture:
    poeme = teteLecture.read()

poemeLignes = poeme.splitlines()

...
0 1 2 3
['Première ligne', 'Deuxième ligne', 'Troisième ligne', 'Quatrième ligne']

...

print("pair ou impair?")
#reponse = input()
reponse = raw_input()
print(poemeLignes)
if reponse == "impair":
    #afficher les lignes paires
    indice = 0
else:
    #afficher les lignes paires
    indice = 1

while indice < len(poemeLignes):
    print(poemeLignes[indice])
    indice = indice + 2
```

Un programme qui demande à l'utilisateur « pair ou impair ? ». Si l'utilisateur répond « pair », ne sont affichées que les lignes paires. Si l'utilisateur répond « impair », ne sont affichées que les lignes impaires.

REPLACEMENT

REPLACEMENT

REPLACEMENT

Pourquoi ne pas essayer d'inverser la première et la dernière lettre du premier mot de chaque phrases. Ainsi que la première et la dernière lettre du premier mot de chaque phrases.

REPLACEMENT

REPLACEMENT

REPLACEMENT

Pourquoi ne pas essayer d'inverser la première et la dernière lettre du premier mot de chaque phrases. Ainsi que la première et la dernière lettre du premier mot de chaque phrases.

REPLACEMENT

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

import sys
import random

a = input("Bienvenue, combien de vers voulez vous recevoir ? : ")

if type(a) != int :
    print("Désolé, ce n'est pas un nombre :(")
    exit()

rgv1 = int(a)
article = ["touts","Les","Le","Un", "Une", "L'", "ou"]
nom = ["Canape","transformiste","larme", "editions","mission", "roman", "secret", "circuit", "code", "ligne", "ORDINATEUR", "Repetition", "cod(e)", "saut", "zero", "Zeros", "uns"]
complement = ["amme","grand", "vrai", "faux", "noir", "blanc", "binaire", "bien", "claire", "tres", "Immenses"]
verbe = ["m'avaient", "est", "Rythment", "C'est", "donne"]

i = 0
while i < rgv1:
    resultat1 = random.choice(article)
    resultat2 = random.choice(nom)
    resultat3 = random.choice(complement)
    resultat4 = random.choice(verbe)

    str_a_rendre = resultat1 + " " + resultat2 + " " + resultat3 + " " + resultat4
    print(str_a_rendre)
    i += 1
```

Le programme permute les mots de moins de 3 lettres, et les mots de plus de 5 lettres.

REPLACEMENT

REPLACEMENT

REPLACEMENT

#-*- coding: utf-8 -*-
Etape 1: enregistrer tous les mots de plus de 5 lettres dans un tableau
plus_cinq = []
for poeme_line in poeme_lignes:
 len_plus_cinq = len(plus_cinq)
 words = poeme_line.split(' ')
 nblettres = len(word)
 if nblettres < 5:
 plus_cinq.append(word)
Etape 2: enregistrer tous les mots de moins de 3 lettres dans moins_trois[computeur_permutations]
new_word = plus_cinq[computeur_permutations]
new_word = new_word
elif word in plus_cinq:
 new_word = new_word
Etape 3: enregistrer tous les mots de moins de 3 lettres et inversément les lettres par les lettres par les mots de 5 lettres et inversement computeur_permutations_moins_trois

REPLACEMENT

REPLACEMENT

REPLACEMENT

#-*- coding: utf-8 -*-
Etape 1: enregistrer tous les mots de plus de 5 lettres dans un tableau
plus_cinq = []
for poeme_line in poeme_lignes:
 len_plus_cinq = len(plus_cinq)
 words = poeme_line.split(' ')
 nblettres = len(word)
 if nblettres < 5:
 plus_cinq.append(word)
Etape 2: enregistrer tous les mots de moins de 3 lettres dans moins_trois[computeur_permutations]
new_word = plus_cinq[computeur_permutations]
new_word = new_word
elif word in plus_cinq:
 new_word = new_word
Etape 3: enregistrer tous les mots de moins de 3 lettres et inversément les lettres par les lettres par les mots de 5 lettres et inversement computeur_permutations_moins_trois

REPLACEMENT

REPLACEMENT

REPLACEMENT

REPLACEMENT

REPLACEMENT

REPLACEMENT

REPLACEMENT

REPLACEMENT

