

trouver le début du code python :

chercher dans les strings de l'image :

```
strings Gorille_Wallpaper.jpg
```

Version rapide :

```
strings Gorille_Wallpaper.jpg | grep import
```

trouver tout le code python

ont modifié un peu la commande grep pour afficher sur une plage de -20lignes à +20lignes

```
strings Gorille_Wallpaper_pasobf.jpg | grep -B 20 -A 20 "import"
```

On peut voir le code python suivant :

```
import string; alph = string.ascii_letters; flag_enc = "109 115 104 110 { 130 111
131 104 111 132 108 122 123 145 104 122 149 121 118 119 149 118 123 }"
def decode_flag(flag):
    lst = flag.split(" ")
    new_flag = ""
    for item in lst:
        if item != '{' and item != '}':
            new_flag += alph[int(item) % 25]
        else:
            new_flag += item
    print(new_flag)
def encode_flag(flag):
    text = ""
    for i in flag:
        if i not in alph:
            text += i + " "
        else:
            text += f"{alph.index(i) + int(flag_enc.split(' ')[2])} + "
    print(text)
decode_flag(flag_enc)
```

Réparer le code python

L'alphabet `string.ascii_letters` contient 52 caractères (26 majuscules et 26 minuscules), donc il faut ajuster les modulo pour éviter les erreurs d'index.

```
import string; alph = string.ascii_letters; flag_enc = "109 115 104 110 { 130 111
131 104 111 132 108 122 123 145 104 122 149 121 118 119 149 118 123 }"
def decode_flag(flag):
    lst = flag.split(" ")
    new_flag = ""
    for item in lst:
        if item != '{' and item != '}':
            new_flag += alph[int(item) % 52]
        else:
            new_flag += item
    print(new_flag)
def encode_flag(flag):
    text = ""
    for i in flag:
        if i not in alph:
            text += i + " "
        else:
            text += f"{alph.index(i) + int(flag_enc.split(' ')[2])} + "
    print(text)
decode_flag(flag_enc)
```

lancer le script python

ont récupère flag{AhBahCestPasTropTot}, ont le remet au format de flag du Midnight :
MCTF{flag{AhBahCestPasTropTot}}