Merci à M^me Renaud pour une grande partie de ce gentil DM.
Vous pouvez compléter et rendre le sujet.

Exo 1 : Bob est puni par son professeur car il bavarde. Il doit donc remplir un fichier de « blablabla... ».

1. Écrire la fonction `repete(chaine,n)` qui prend en entrée une chaine de caractère `chaine` et un entier `n` et renvoie une chaine de caractère contenant `n` fois la chaine fournie en entrée.
   Par exemple, `repete("bla",3)` doit retourner `"blablabla"`

2. Bob souhaite maintenant écrire le résultat dans un fichier. Il a commencé le code suivant.

```
monfichier=open("C://Bureau/mapunition.txt", ... )
monfichier.write(.............)
...
```

   Il hésite sur le mode d'ouverture du fichier entre `r`, `w` et `a`. Pouvez vous lui rappeler le cours ?
   - `r` :
   - `w` :
   - `a` :

   Complétez le code afin que le fichier contienne « blablablablabla »

Exo 2 :

1. Écrire une fonction `test_div_7(n)` qui renvoie `True` si l'entier `n` fourni en paramètre d'entrée est divisible par 7 et `False` sinon.

2. Soit f la fonction qui a un nombre m fourni en entrée (en base 10) associe le nombre obtenu comme suit :
   - on enlève à l'écriture décimale de m son chiffre des unités
   - on lui retranche deux fois le chiffre des unités de m

   Par exemple $f(31976) = 3197 - 2 \times 6 = 3185$
   De même, $f(3185) = 318 - 2 \times 5 = 308$.

   (a) Écrire en python la fonction f

   (b) On appelle f dans le code de la fonction suivante :

```
def mystere(m):
    while m>99:
        m=f(m)
    return m
```

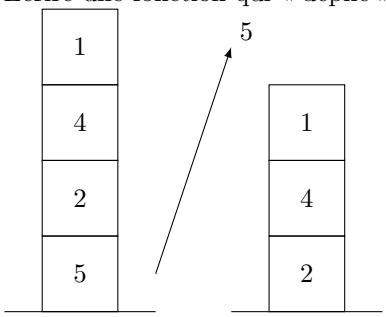
   Quel est le variant de boucle ? Justifier brièvement que la boucle termine.

   (c) Pour savoir si le nombre m est divisible par 7, il suffit de déterminer si `mystere(m)` est dans la liste des multiples de 7 inférieurs à 100. Écrire une fonction `test_div_7bis(n)` utilisant cette méthode.

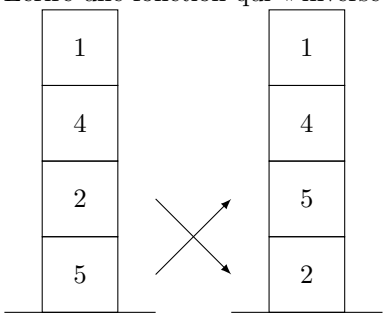
Exo 3 : Réviser les piles ! Lors de l'utilisation des piles, on pourra utiliser les 6 fonctions suivantes :

— `creer_pile()` qui ne prend pas d'argument et renvoie une pile
— `est_vide(p)` qui prend en argument une pile et renvoie un booléen
— `empiler(p,e)` qui prend en argument une pile et un élément et ne renvoie rien
— `depiler(p)` qui prend en argument une pile et renvoie un élément
— `sommet(p)` qui prend en argument une pile et renvoie un élément
— `taille(p)` qui prend en argument une pile et renvoie un entier

1. Écrire une fonction qui « dépile » l'élément à la base de la pile.



2. Écrire une fonction qui « inverse » les deux éléments à la base de la pile.



Exo 4 :

On définit la fonction `mystere` suivante prenant en paramètres d'entrée a, entier supérieur à 1 et b entier supérieur ou égal à 2.

```
def mystere(a,b):
    if a < b :
        return 0
    else:
        return 1 + mystere( a//b , b )
```

1. Que renvoie l'appel de `mystere(1001,10)` ?
2. Exprimer ce que renvoie `mystere` à partir des fonctions usuelles (partie entière, logarithme ...).
3. Que se passe-t-il si $b = 1$?
4. Préciser, en justifiant, le coût d'un appel selon a et b.