





Dans cet exercice, on considère une pile d'entiers positifs. On suppose que les quatre fonctions suivantes ont été programmées préalablement en langage Python :

empiler(P, e) : ajoute l'élément e sur la pile P ;
 depiler(P) : enlève le sommet de la pile P et retourne la valeur de ce sommet ;
 est_vide(P) : retourne True si la pile est vide et False sinon ;
 creer_pile() : retourne une pile vide.

Dans cet exercice, seule l'utilisation de ces quatre fonctions sur la structure de données pile est autorisée.

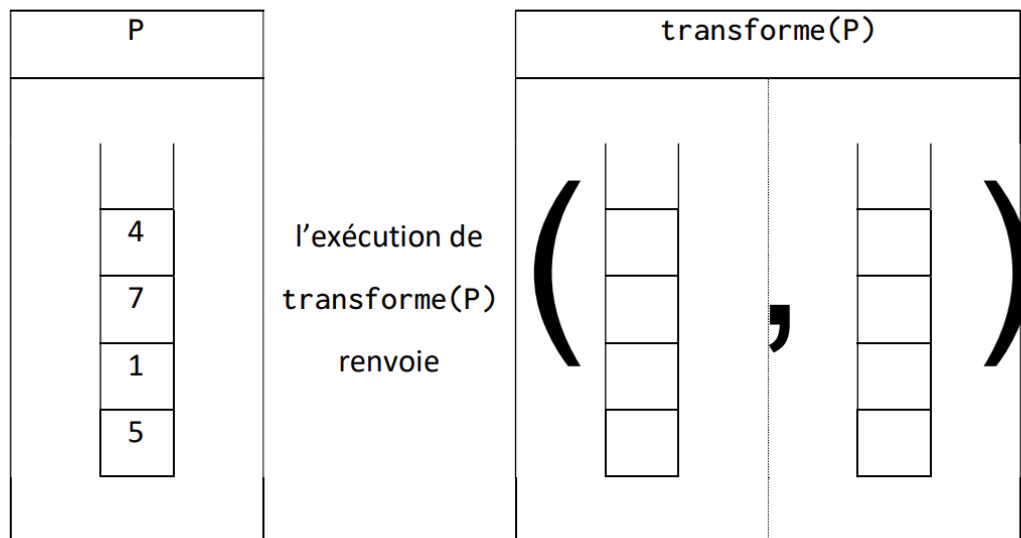
- 1. Recopier** le schéma ci-dessous et le compléter sur votre copie en exécutant les appels de fonctions donnés. On écrira ce que renvoie la fonction utilisée dans chaque cas, et on indiquera None si la fonction ne retourne aucune valeur.

	Etape 0 Pile d'origine P	Etape 1 empiler(P,8)	Etape 2 depiler(P)	Etape 3 est_vide(P)
				
Retour de la fonction	

- 2.** On propose la fonction ci-dessous, qui prend en argument une pile P et renvoie un couple de piles :

```
def transforme(P) :
    Q = creer_pile()
    while not est_vide(P) :
        v = depiler(P)
        empiler(Q,v)
    return (P,Q)
```

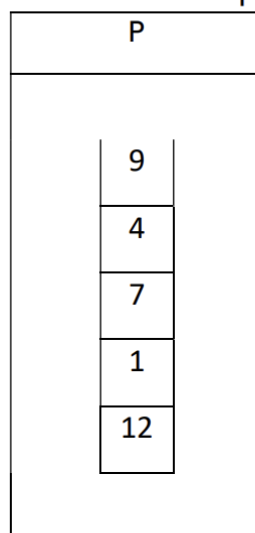
Recopier et compléter sur votre copie le document ci-dessous



3. Ecrire une fonction en langage Python `maximum(P)` recevant une pile `P` comme argument et qui renvoie la valeur maximale de cette pile. On ne s'interdit pas qu'après exécution de la fonction, la pile soit vide.

4.

On souhaite connaître le nombre d'éléments d'une pile à l'aide de la fonction `taille(P)`



`taille(P)` retournera donc l'entier 5

A la fin de l'exécution de la fonction, la pile `P` ne devra pas être vide.

4.a. Proposer une stratégie écrite en langage naturel et/ou expliquée à l'aide de schémas, qui permette de mettre en place une telle fonction.

4.b. Donner le code Python de cette fonction `taille(P)` (on pourra utiliser les cinq fonctions déjà programmées).