

Les trois exercices sont à traiter.

Rédiger chacun d'entre eux sur une copie double SVP (donc rendre trois copies). Merci.

Le sujet comporte 6 pages : vérifier qu'il est bien complet avant de commencer.

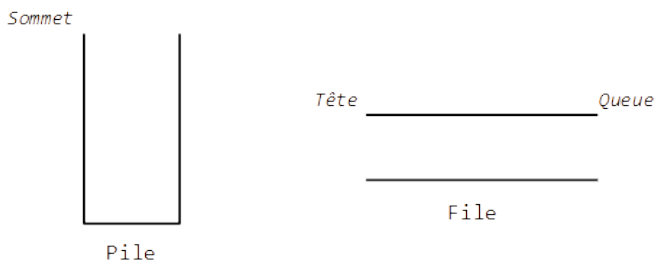
Exercice 1 - Structures abstraites de données et programmation en général.

On dispose des structures de données suivantes :

- une pile avec les fonctions primitives suivantes : `creerPile()`, `estVidePile()`, `depiler()`, `empiler()` `lireSommet()` et `hauteur()`
- une file avec les fonctions primitives suivantes : `creerFile()`, `estVideFile()`, `defiler()`, `enfiler` et `longueur()`

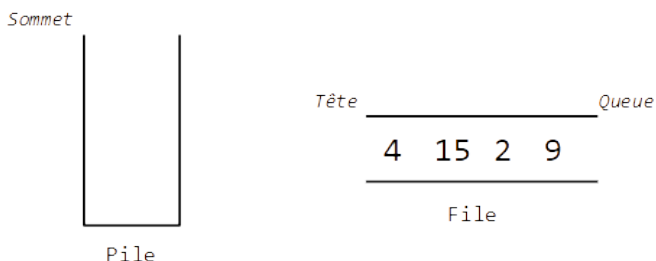
On crée une file et une pile vides.

On schématise la pile et la file de la manière suivante :



Question 1 - Recopier sur la copie le schéma ci-dessus et le compléter afin d'indiquer l'état de la pile et de la file après avoir exécuté les opérations suivantes : `enfiler(12)`, `enfiler(4)`, `enfiler(7)`, `empiler(defiler())`, `empiler(3)`.

On crée une nouvelle file et une nouvelle pile vides. On enfile les valeurs suivantes : 4, 15, 2 et 9.



On veut exécuter l'algorithme suivant :

début

tant que la file n'est pas vide

 défiler

tant que la pile n'est pas vide

 et tant que la valeur défilée est supérieure à la valeur du sommet de la pile

```

    dépiler et enfiler la valeur défilée
  fin tant que
  empiler la valeur défilée
fin tant que
fin

```

Question 2a - Recopier sur la copie le schéma ci-dessus et le modifier afin d'indiquer l'état de la pile et de la file après avoir effectué deux cycles de la boucle principale (tant que la file n'est pas vide...) de cet algorithme.

Question 2b - Recopier sur la copie le schéma ci-dessus et le modifier afin d'indiquer l'état de la pile et de la file après avoir effectué l'intégralité de cet algorithme.

Question 3 - Indiquer ce que réalise cet algorithme et préciser quel en est le coût.

On implémente en programmation impérative cet algorithme en langage Python.

```

1  from module_Pile_File import *
2
3  def algo(p, f):
4
5      while estVideFile(f) is False:
6
7          x = defile(f)
8
9          while estVidePile(p) is False and lireSommet(p) < x:
10             z = depile(p)
11             enfile(f, z)
12
13             empile(p, x)
14
15         return p, f
16
17     # Programme Principal
18     p = creerPile()
19     f = creerFile()
20     enfile(f, 4)
21     enfile(f, 15)
22     enfile(f, 2)
23     enfile(f, 9)
24     p, f = algo(p, f)

```

```

1  from module_Pile_File import *
2
3  def algo1(p, f):
4      if estVideFile(f):
5          return ...
6
7      x = defile(f)
8      algo2(p, f, x)
9      return ...
10
11
12  def algo2(p, f, x):
13      if estVidePile(p) ... lireSommet(p) ... x:
14          empile(p, x)
15          return ...
16
17      z = depile(p)
18      enfile(f, z)
19      return ...
20
21
22     p = creerPile()
23     f = creerFile()
24     enfile(f, 4)
25     enfile(f, 15)
26     enfile(f, 2)
27     enfile(f, 9)
28     p, f = algo1(p, f)
29

```

On veut écrire une version récursive de la fonction algo sous la forme de deux fonctions algo1 et algo2, chacune d'elles correspondant à l'une des boucles non bornées de la fonction algo (voir ci contre).

Question 4 - Recopier sur la copie les lignes 5, 9, 13, 15 et 19 en les complétant de façon à ce que les deux fonctions puissent s'exécuter correctement.