

EXERCICE 1 (4 points)

Cet exercice porte sur la notion de file et sur la programmation de base en Python.

On rappelle qu'une file est une structure de données abstraite fondée sur le principe « premier arrivé, premier sorti ».

On munit la structure de données File des fonctions primitives suivantes.

Structure de données : File

Utilise : Booleen, Element

Opérations

- `creer_file` : $\emptyset \rightarrow \text{File}$

`creer_file()` renvoie une file vide.

- `est_vide` : $\text{File} \rightarrow \text{Booleen}$

`est_vide(f)` renvoie `True` si la file `f` est vide et `False` sinon

- `enfiler` : $\text{File}, \text{Element} \rightarrow \text{Rien}$

`enfiler(f, e)` ajoute l'élément `e` en queue de la file `f`

- `defiler` : $\text{File} \rightarrow \text{Element}$

`defiler(f)` renvoie l'élément qui est en tête de la file `f` et le retire de celle-ci.

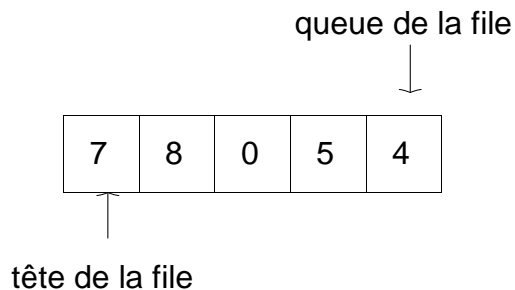


Figure 1 : une file

Partie A :

1. On suppose dans cette question que la file `f` est dans l'état de la *figure 1*.

Quel est l'état de la file `f` après l'exécution des instructions suivantes?

```
1  for k in range(3) :
2      x = defiler(f)
3      enfiler(f, x)
```

2. On suppose dans cette question que la file *f* est dans l'état de la *figure 1*.

```
1 k = 0
2 g = creer_file()
3 while not est_vide(f) :
4     x = defiler(f)
5     k = k + 1
6     enfiler(g, x)
7 while not est_vide(g) :
8     x = defiler(g)
9     enfiler(f, x)
```

Après l'exécution des instructions précédentes :

- (a) Quel est l'état de la file *f* ?
- (b) Quelle est la valeur de *k* ? Que représente la variable *k* ?

Partie B :

Une imprimante a une file d'attente d'impression dans laquelle sont placés les fichiers à imprimer. On peut modéliser cette file d'attente par une file dans laquelle les éléments sont des numéros permettant d'identifier les fichiers à imprimer. Ainsi tous les éléments dans la file sont différents deux à deux.

3. Dans la gestion de l'imprimante, on a la possibilité de supprimer un fichier de la file d'attente d'impression.

Voici le code incomplet de la fonction *supprimer* qui a pour paramètres une file *f* et un élément *e* présent une seule fois dans *f* et qui retire l'élément de la file. La fonction ne renvoie rien.

```
1 def supprimer(f, e) :
2     g = creer_file()
3     while not est_vide(f) :
4         x = defiler(f)
5     #ligne à compléter
6     #ligne à compléter
7     while not est_vide(g) :
8     #ligne à compléter
9         enfiler(f, x)
```

Recopier et compléter le code de la fonction.

4. Dans la gestion de l'imprimante, une option permet d'imprimer un fichier en priorité. Ainsi le fichier est placé en tête de la file d'attente d'impression.

En modifiant la fonction *supprimer*, écrire une fonction *placer_en_priorite* qui a pour paramètres une file *f* et un élément *e* et qui déplace cet élément en tête de file. On suppose que l'élément *e* est déjà présent dans la file.