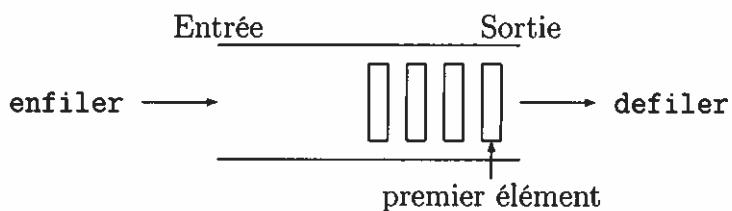


## Exercice 5 (4 points).

Cet exercice porte sur les files, les tableaux et les algorithmes associés.

L'objectif de cet exercice est de travailler sur les températures relevées par une station météorologique. Les données sont enregistrées une fois par jour, à la même heure, et traitées dans l'ordre dans lequel elles arrivent.

On choisit d'utiliser une file : une file est une structure de données abstraite fondée sur le principe « premier arrivé, premier sorti ».



On munit la structure de données File des quatre fonctions primitives définies ci-dessous :

### Structure de données abstraite : File

Utilise : Élément, Booléen

#### Opérations :

- **creer\_file\_vide** :  $\emptyset \rightarrow \text{File}$   
`creer_file_vide()` renvoie une file vide
- **est\_vide** :  $\text{File} \rightarrow \text{Booléen}$   
`est_vide(F)` renvoie `True` si la file F est vide, `False` sinon
- **enfiler** :  $\text{File}, \text{Elément} \rightarrow \emptyset$   
`enfiler(F, element)` ajoute `element` en entrée de la file F
- **defiler** :  $\text{File} \rightarrow \text{Elément}$   
`defiler(F)` renvoie l'élément en sortie de la file F (premier élément) en le retirant de la file F

1. Les températures relevées ont été 15, puis 17, puis 14.

(a) Parmi les quatre propositions suivantes, indiquer celle qui représente correctement cette file :

- |                        |                    |                           |
|------------------------|--------------------|---------------------------|
|                        | Entrée      Sortie |                           |
| <b>Proposition 1 :</b> | <u>15 17 14</u>    | Le premier élément est 14 |
|                        | Entrée      Sortie |                           |
| <b>Proposition 2 :</b> | <u>14 17 15</u>    | Le premier élément est 15 |
|                        | Entrée      Sortie |                           |
| <b>Proposition 3 :</b> | <u>15 17 14</u>    | Le premier élément est 15 |
|                        | Entrée      Sortie |                           |
| <b>Proposition 4 :</b> | <u>14 17 15</u>    | Le premier élément est 14 |

(b) En utilisant les fonctions primitives précédentes, donner les instructions permettant de créer cette file.

2. On appelle longueur d'une file le nombre d'éléments qu'elle contient.

La fonction `longueur_file` prend en paramètre une file F et renvoie sa longueur n.

Après appel de cette fonction, la file F doit avoir retrouvé son état d'origine.

**Exemple :**

Si F = 10 10 12 12 alors `longueur_file(F)` vaut 4.

Recopier et compléter le programme Python suivant, implémentant la fonction `longueur_file`.

Dans le code de la fonction, les trois points (...) peuvent correspondre à une ou plusieurs lignes de programme.

```

1 def longueur_file(F):
2     """File -> Int"""
3     G = creer_file_vide() # file temporaire
4     n = 0 # initialisation du nombre d'elements
5     while not(est_vide(F)):
6         ...
7         while not(est_vide(G)): # reconstruction de la file initiale
8             ...
9     return ...

```

3. On s'intéresse à la variation de température d'un jour sur l'autre.

Par exemple, lorsque les températures relevées sont dans l'ordre d'arrivée 15, 17 et 14, les variations sont 2 et -3.

Recopier et compléter le programme Python implémentant la fonction `variations` qui prend en paramètre une file non vide F et qui renvoie le tableau tab contenant les variations successives, ou un tableau vide si la file F ne contient qu'une seule température. Il n'est pas demandé ici que la file F retrouve son état d'origine après appel de la fonction `variations`.

**Exemple :** si F est la file qui contient dans l'ordre des relevés les valeurs 15, 17 et 14, `variations(F)` vaut [2, -3].

Dans le code de la fonction, les trois points (...) peuvent correspondre à une ou plusieurs lignes de programme.

```

1 def variations(F):
2     """File -> Tableau"""
3     taille = longueur_file(F)
4     if taille == 1:
5         ...
6     else:
7         tab = [0 for k in range(taille - 1)]
8         element1 = defiler(F)
9         for i in range(taille - 1):
10            element2 = defiler(F)
11            ...
12     return ...

```

4. Écrire une fonction `nombre_baisse` qui prend en paramètre un tableau tab, non vide, des variations des températures et qui renvoie un p-uplet contenant le nombre de jours où la

température a baissé par rapport au jour précédent (soit le nombre de valeurs strictement négatives de `tab`), ainsi que la baisse journalière la plus importante (soit la valeur minimale de `tab`).

S'il n'y a aucune baisse (toutes les valeurs de `tab` sont positives), la fonction renvoie le p-uplet  $(0,0)$ .

**Exemple 1 :** `nombre_baisses([1, -4, 2, -1, 3])` vaut  $(2, -4)$ .

**Exemple 2 :** `nombre_baisses([1, 5, 3, 1])` vaut  $(0,0)$ .