Lycée du	Les Types abstraits de données – Les files		
Rempart	Evaluation		
Nom, Prénom :	Classe: Dat	e :	
Appréciation :	Not	e :	

1. Organisation

Sauf indication, les réponses sont à rédiger dans les emplacements prévus, à la suite des questions.

2. Les files : une implémentation par tableau

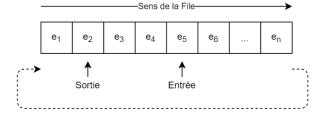
On se propose de réaliser l'implémentation d'une file par un tableau de taille fixe via la programmation orientée objet.

Le code correspondant est celui fourni en Annexe 1. *Remarque* : ce code est incomplet.

Dans une première partie on étudiera le fonctionnement de cette implémentation. Puis, dans la seconde, on en complétera le code.

Fonctionnement

La file utilise un tableau de taille n, dont l'indexage commence à zéro, comme illustré cicontre.



L'entrée (tête) de la file correspond à l'index « entrée »

La sortie (queue) de la file correspond à l'index « sortie ».

Lors d'une opération enfiler, l'élément ajouté à la file est mémorisé dans la case pointée par l'index entrée puis, la valeur de cet index est augmentée de 1.

Lors d'une opération défiler, l'élément obtenu est celui mémorisé dans la case pointée par l'index sortie puis, la valeur de cet index est augmentée de 1.

Lorsque la valeur d'un des index (entrée ou sortie) atteint la taille n du tableau, celle-ci revient à zéro. Ainsi, cette valeur continue à croitre, toujours dans le même sens.

Lorsque les valeurs des index coïncident (c'est-à-dire qu'ils ont la même valeur), cela signifie que :

- Soit la file est vide ;
- Soit la file est pleine.

Lycée du	Les paradigmes de programmation	
Rempart	Evaluation	2/5

Ci-dessous, quelques exemples de fonctionnement. <u>Remarque</u>: les index entrée et sortie sont nommés par leurs initiales.

Opération	Etat de la file	
Création de la file (vide) :		
f = File()	None None None None None None None	
	↑ ↑ s e	
Enfiler(« a »):	Sens de la File	
f.enfiler("a")	"a" None None None None None None	
	↑ ↑ s e	
Enfiler(« b »):	Sens de la File	
f.enfiler("b")	"a" "b" None None None None None	
	↑	
Défiler un élément :	Sens de la File	
<pre>element = f.defiler()</pre>	"a" "b" None None None None None	
→ « a » obtenu.	↑ ↑ s e	
Enfiler(« c ») :	Sens de la File	
<pre>f.enfiler("c")</pre>	"a" "b" "c" None None None None	
	↑ ↑ s e	
Défiler un élément :	Sens de la File	
<pre>element = f.defiler()</pre>	"a" "b" "c" None None None None	
→ « b » obtenu.	↑ ↑ s e	
Défiler un élément :	Sens de la File	
<pre>element = f.defiler()</pre>	"a" "b" "c" None None None None	
→ « c » obtenu.	↑ ↑ s e	

/1	1)	A partir du tableau de fonctionnement ci-dessus, donner le nombre de cas où la file es
		vide.

	Rempart	Evaluation	3/5
/2	- Le nom	la classe File (donné en Annexe 1), donner : de l'attribut qui sert à mémoriser le tableau ; e de ce tableau.	
/2		e la classe File (donné en Annexe 1), expliquer le principe utilisé es états vide et pleine de la file.	pour
/1	4) Donner le n	nombre de lignes où est utilisé la taille du tableau.	
/2,5		i l'utilisation de la taille du tableau tel quel dans le code est un l	usage
Bonus +1	Pour répondre réutilisation et	Justifier votre réponse. à cette question, vous devrez considérer les principes de modularit de modification de code. Si non, proposez une modification simple er le principe sans réécrire la classe.	

Les paradigmes de programmation

 $\mathsf{T}^\mathsf{e}\,\mathsf{NSI}$

Lycée du

	Rempart	Ev	valuation	4/5
6)) Donner la p	récondition pour qu'un élément	puisse être ajouté dans	la file.
Со	mpléter la cla	asse		
7)) Donner la p	récondition pour qu'un élément	puisse être retiré de la f	ile.
5 8)) Lorsque la p	orécondition pour qu'un élément	puisse être retiré de la f	ile <u>n'est pas</u> vérifiée,
	donner la valeur (ou l'objet) que devrait renvoyer la méthode defiler()? Justifier votr réponse.			
9)	9) <u>Dans l'annexe 1</u> , Rédiger la méthode defiler() d'après les spécifications données cidessus. Votre code devra tenir compte de la réponse apportée à la question 5). Cependant, si vous n'y avez pas répondu, aucune pénalité ne vous sera attribuée.			
10	10)Dans le tableau ci-dessous, donner les complexités de cette implémentation du type abstrait de donnée file et conclure sur ses performances.			
С	réation :		Enfiler:	
Т	est si File vide	2:	Défiler :	
<u>C</u>	Conclusion :			

Les paradigmes de programmation

T^e NSI

Lycée du

Lycée du	Les paradigmes de programmation	
Rempart	Evaluation	5/5

ANNEXE 1 - Code de la classe File.

```
class File:
def __init__(self):
    self.file = [None for i in range(100)]
    self.entree = 0 # Index de la tête de la file
    self.sortie = 0 # Index de la queue de la file
    self.longueur = 0 # Nombre d'éléments contenus dans la file
def est_vide(self) -> bool:
    """ Renvoie True si la file est vide """
    return self.longueur == 0
def est_pleine(self) -> bool:
    """ Renvoie True si la file est pleine """
    return self.longueur == 100
def enfiler(self, donnee) -> bool:
    """ Enfile un élément dans la file. renvoie True si résussite, False sinon """
    if self.est_pleine():
        return False
    else:
        self.file[self.entree] = donnee
        self.longueur += 1
        self.entree +=1
        if self.entree == 100:
             self.entree = 0
        return True
def defiler(self):
    """ Défile un élément de la file. Renvoie cet élément """
```