

**PARTIEL 1**

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant imprimé (et celui-ci seulement) et les notes manuscrites (pas de photocopie) comprenant le nom de l'étudiant sur chaque page.

Il est conseillé d'écrire les exercices en langage C++. L'écriture en Java complique les entrées-sorties et l'écriture en langage C allonge le programme (ce qui ne rapportera pas de point supplémentaire dans un cas comme dans l'autre).

Exercice unique.- **File d'attente circulaire bornée**

*Dans certains cas, par exemple le nombre de clients en attente de connexion sur un serveur ou le nombre de trames à transmettre à une carte réseau, la file d'attente ne doit pas contenir plus d'un certain nombre d'items. Dans ce cas la file d'attente est conçue comme un tableau avec deux index : **head** pointant sur le prochain item à enlever et **tail** pointant sur le prochain item à ajouter.*

head	1
	2
	3
	4
tail	

*On parle de file d'attente circulaire car, lorsque **tail** (puis **head**) atteignent la dernière ligne, la fois suivante ils passent à la première ligne, comme le montre la figure ci-dessous :*

	4
tail	
head	1
	2
	3

- 1°) a) Quelles seront les positions de **head** et de **tail** au départ ?
  - b) Comment reconnaît-on que la file a atteint sa capacité maximum, autrement dit qu'elle est pleine ?
  - c) Comment reconnaît-on que la file est vide ?

- 2°) Implémenter une file d'attente circulaire d'entiers **file** dont les attributs sont la capacité (8 items dans notre exemple), un tableau d'entiers, deux index **tail** et **head** et la longueur effective de la file d'attente (4 dans notre exemple) et comme méthodes un constructeur prenant un argument entier (la capacité), **estVide()**, **estPleine()**, **placer()**, **enlever()** et **longueur()** qui renvoie la longueur effective de la file d'attente.

[ *On fera attention de ne pas placer un élément lorsque la file d'attente est pleine ou d'enlever lorsqu'elle est vide. Aucune des méthodes ne sera en ligne.* ]

- 3°) Ajouter une nouvelle méthode booléenne **isIn()** prenant un argument entier qui prend la valeur vraie si l'argument est dans la file d'attente.

[ *Ceci est le début pour implémenter une structure de données abstraites concernant les ensembles bornés d'entiers.* ]

- 4°) Écrire un programme de test.