VARI1 NFP135 Exercices de révision

Exercice 1 :

Une file est une structure de données dont les éléments sont accédés selon le mode FIFO (First In First Out). Cela signifie que seul l'élément le plus ancien dans la file peut en être extrait.
Une file est gérée à l'aide :

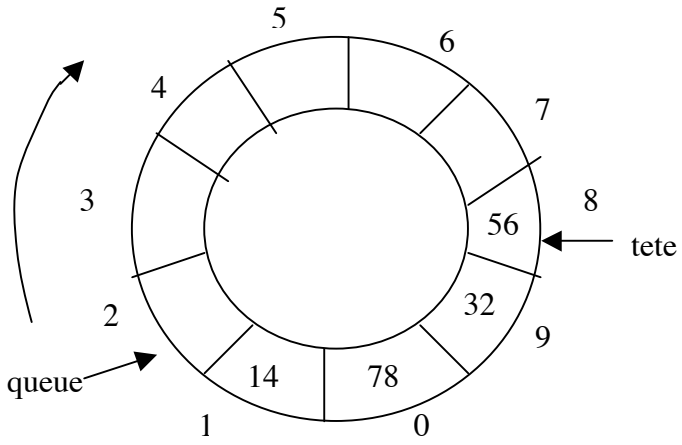
- d'un pointeur (ou index) de tête à partir duquel un élément peut être extrait
- d'un pointeur (ou index) de queue à partir duquel un élément peut être ajouté

Les opérations principales sont :

`enfiler`, qui ajoute un élément en queue de file
`defiler`, qui extrait un élément de la tête de file
`sommet`, qui retourne l'élément situé en tête sans l'enlever de la file
`estvide`, qui retourne vrai si la file est vide
`estpleine`, qui retourne vrai si la file est pleine
`put` qui affiche le contenu de la file

L'interface du paquetage `files` qui suit, exporte le type `File` dont les éléments sont des entiers positifs. Nous avons choisi de représenter une file par un tableau géré circulairement.

Dans cet exemple, on voit que l'indice de queue indique la place du prochain élément à insérer alors que l'indice de tête indique la place du prochain élément à retirer de la file. La valeur de ces 2 indices évoluent modulo la taille du tableau.




```
package files is
   fileVide  : exception;
   filePleine: exception;
   type File is private;

   procedure enfiler  ( t: ? Positive; f: ? File );
   procedure defiler  ( t: ? Positive ; f: ? File );
   function  sommet   ( f: File ) return Positive ;
   function  estvide  ( f: File ) return Boolean;
   function  estpleine( f: File ) return Boolean;
   procedure put      ( f: ? File );
private
-- MAX représente le nombre maximum d'éléments que peut contenir la file
   MAX : constant Natural :=30 ;
```

```
   type TabFile is array( Natural range 0..MAX-1 ) of Positive;
-- nbElements représente le nombre d'éléments présents à un instant donné
dans la file.
   type File is
      record
         tab        : TabFile;
         tete       : Natural:=0;
         queue      : Natural:=0;
         nbElements : Natural range 0..MAX :=0;
      end record;
end files;
```

Question 1 (1 point)

Dans l'interface du paquetage `files` ci-dessus, remplacer les points d'interrogation par les modes de passage de paramètres corrects.

Question 2 (3 points)

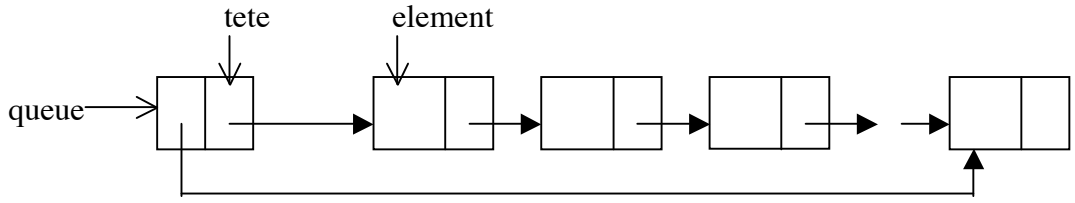
Définir le corps de la procédure `defiler`.

Question 3 (3 points)

Définir le corps de la procédure `enfiler`.

Question 4 (3 points)

On se propose de changer la représentation physique d'une file. On choisit de la représenter sous la forme d'une liste chaînée conforme au schéma suivant :



Modifier la partie privée de l'interface du paquetage `files` pour prendre en compte cette nouvelle représentation. On limitera le nombre d'éléments de la file à `MAX`.

Question 5 (3 points)

Définir le corps de la procédure `enfiler`.

Question 6 (1 point)

Transformer la partie publique de l'interface du paquetage `files` de manière à le généraliser pour n'importe quel type d'élément.

Exercice 2 : jeu des cinq erreurs

Le programme suivant comporte 5 erreurs de compilation.
Décrivez les erreurs en donnant le numéro de la ligne et la cause de chacune d'entre elles. Il est possible qu'une même erreur ait plusieurs occurrences dans le texte. Dans ce cas, on ne la considère que comme une unique erreur.

```
1 with Ada.Float_Text_io; use Ada.Float_Text_io;
2 with Ada.Text_io;

3 procedure cinq_erreurs is
4  type Vecteur is array( Positive range <> ) of Float;
5  procedure inserer( x:in Float;a:in Vecteur;i:in Positive ) is
6     erreur:exception;
7   begin
8     if i<=a'last and i>=a'first
9     then
10        a(i):=x;
11    else
12       raise Erreur;
13    end if;
14 end inserer;

15 procedure put( a:out Vecteur ) is
16 begin
17  put( "(" );
18  for i in a'range loop
19    put( a(i),2,1,0 );
20    put( "," );
21  end loop;
22  put(")");
23 end put;

24 v:Vecteur:=( 3.5 ,7.8 ,9 ,2.2 ,6.3 ,0.7 ,7.6 );

25 begin
26  for i in v'range loop
27     inserer( 5.8,v,i );
28  end loop;
29   put(v);
30 exception
31   when erreur => put( "vous etes en dehors des bornes du tableau!" );
32 end cinq_erreurs;
```