

Solutionnaire pour les exercices sur les listes chaînées et les files

1. Définition récursive d'une liste simplement chaînée (après vérification avec Sylvie, il vaut mieux la définir avec un algorithme, en oubliant le formalisme mathématique.)

Une liste est soit vide soit un nœud (ou cellule) suivi d'une liste. Un nœud est composé d'un Objet suivi d'une référence vers un Nœud .

Nœud { Objet (qui peut être une référence) Référence vers un Nœud (Suivant) }	Nouvelle Liste (Nœud, Liste) { Si Liste = NULL Faire Liste := Nœud; Retourner Liste Sinon Liste = Liste. Suivant Fin Si Nouvelle Liste (Nœud, Liste) }
--	---

2. Voici une méthode pour insérer un élément au début d'une liste simplement chaînée. On garde le pointeur de la Tête dans un pointeur temporaire. On fait ensuite pointer le début de la liste au nouveau nœud à insérer. Puis on met le suivant de cet objet au pointeur temporaire gardé.

Temporaire := Tête

Tête := Nouveau

Tête. Suivant := Temporaire

3. Programme récursif qui supprime le dernier élément d'une liste simplement chaînée.

SupprimerDernier (Courant, soit un pointeur sur un nœud de la liste) {

Si Courant. Suivant n'est pas vide **faire**

 Courant := Courant. Suivant

 SupprimerDernier (Courant)

Sinon

 Effacer le contenu de Courant

 Mettre Courant à NULL

Fin si }

On pourrait remplacer les lignes « Effacer le contenu de Courant » et « Mettre Courant à NULL » par « Dépiler (Courant) ».

4. Voici un algorithme récursif pour concaténer deux listes doublement chaînées L et M en une seule liste L' contenant tous les nœuds de L puis tous les nœuds de M.

```

Concaténer (Courant = Début de L, Début de M)
{
    Si Courant = Fin de L Faire
        Courant. Suivant := Début de M
        Début de M. Précédent := Courant
        Retourner L (qui contiendra les deux listes concaténées)
    Fin si
    Concaténer (Courant, Début de M)
}

```

5. Voici la sortie de la série d'opérations :

Ajouter (5)	<table border="1"><tr><td>5</td></tr></table>	5			
5					
Ajouter (3)	<table border="1"><tr><td>5</td><td>3</td></tr></table>	5	3		
5	3				
Enlever ()	Enlève et retourne 5				
Ajouter (2)	<table border="1"><tr><td>3</td><td>2</td></tr></table>	3	2		
3	2				
Ajouter (8)	<table border="1"><tr><td>3</td><td>2</td><td>8</td></tr></table>	3	2	8	
3	2	8			
Enlever ()	Enlève et retourne 3				
Enlever ()	Enlève et retourne 2				
Ajouter (9)	<table border="1"><tr><td>8</td><td>9</td></tr></table>	8	9		
8	9				
Ajouter (1)	<table border="1"><tr><td>8</td><td>9</td><td>1</td></tr></table>	8	9	1	
8	9	1			
Enlever ()	Enlève et retourne 8				
Ajouter (7)	<table border="1"><tr><td>9</td><td>1</td><td>7</td></tr></table>	9	1	7	
9	1	7			
Ajouter (6)	<table border="1"><tr><td>9</td><td>1</td><td>7</td><td>6</td></tr></table>	9	1	7	6
9	1	7	6		
Enlever ()	Enlève et retourne 9				
Enlever ()	Enlève et retourne 1				
Ajouter (4)	<table border="1"><tr><td>7</td><td>6</td><td>4</td></tr></table>	7	6	4	
7	6	4			
Enlever ()	Enlève et retourne 7				
Enlever ()	Enlève et retourne 6				

6. Voici la sortie de la série d'opérations :

AjouterDébut (3)	3
AjouterFin (8)	3 8
AjouterFin (9)	3 8 9
AjouterDébut (5)	5 3 8 9
EnleverDébut ()	Enlève et retourne 5
EnleverFin ()	Enlève et retourne 9
AjouterFin (7)	3 8
EnleverDébut ()	Enlève et retourne 3

7. On performe 32 opérations ajouter(objet), 10 opérations devant() et 15 opérations enlever() parmi lesquelles 5 ont généré des ExceptionFileVide. Donc **10 opérations enlever()** ont retiré des éléments. **32 ajouts – 10 retraits = 22 objets restants. Taille courante = 22.**

8. Pour implémenter une pile avec deux files, il faudrait utiliser une file pour les opérations d'empilement et l'autre pour les opérations de dépilement. Ainsi, lors de l'insertion d'un objet, on l'enfile dans la première file. Par la suite, on le copie dans la deuxième file, que l'on inverse, puisqu'elle devra être de sens contraire pour que le dernier objet ajouté soit le premier objet enlevé. Ainsi toutes les opérations se feront en temps linéaire.

9. Voici les valeurs renvoyées par les opérations d'Enlever

	E	A	S	*	Y	*	Q	U	E	*	*	*	S	T	*	*	*	I	O	*	N	*	*	*
sortie				E		A				S	Y	Q			U	E	S			T		I	O	N

10. Voici les valeurs renvoyées par les opérations d'enleverDébut et d'enleverFin.

	E	A	s	+	Y	+	Q	U	E	*	*	*	s	t	+	*	+	I	O	*	n	+	+	*
sortie				A		Y				s	e	q			E	t	U			O		I	S	n

11. Avec la séquence EasY...

- (i) E+sa**Y+
- (ii) YasE : c'est impossible, puisque si je veux enlever le Y, je dois l'avoir ajouté, c'est-à-dire que j'ai déjà ajouté a et s à la fin de la queue, et que si je veux les enlever de la fin de la queue, ils sortiront dans l'ordre « sa » et non « as ».
- (iii) Ea*sY+*+
- (iv) Ea*s*Y++