

Les structures de données linéaires

Site Internet :
www.gecif.net

Type de document :
Cours

Intercalaire :

Date :

Les structures de données linéaires se décomposent en 3 types :

- les **listes**
- les **pires**
- les **files**

Ces structures de données sont rarement prévues nativement dans les langages de programmation comme type de données : pour pouvoir les utiliser il faudra alors avant tout les programmer.

Une structure de données possède un ensemble de routines (procédures ou fonctions) permettant de manipuler les données enregistrées (ajout, suppression, ou modification d'une donnée). Cet ensemble de routines est appelé **interface**.

I - Les listes

La structure de données linéaire appelé « **liste** » est similaire à un **tableau** permettant de regrouper des données où chaque élément est accessible par un **index** numérique. Une liste est une structure de données dont l'accès est séquentiel (les données sont traitées une seule à la fois) et direct (on peut accéder à tout instant à n'importe quelle donnée sans toucher aux autres).

Voici les opérations qui peuvent être effectuées sur une liste (**interface** d'une liste) :

- CREER_LISTE_VIDE() : crée et retourne une liste vide
- INSERER(L,e,i) : insère l'élément e à la position i dans la liste L (en décalant tous les éléments en position supérieure à i)
- SUPPRIMER(L,i) : supprime de la liste L l'élément qui est en position i (en décalant tous les éléments en position supérieure à i)
- RECHERCHER(L,e) : recherche l'élément e dans la liste L et retourne son index
- LIRE(L,i) : retourne l'élément qui est en position i dans la liste L
- MODIFIER(L,i,e) : l'élément à la position i dans la liste L est remplacée par l'élément e
- TAILLE(L) : retourne le nombre d'éléments présents dans la liste L (taille ou longueur de la liste)

Remarques :

- dans une liste il n'y a jamais de « trou », c'est-à-dire d'emplacement vide entre deux emplacements pleins
- la position du premier élément d'une liste est en principe numéroté 1 (ou plus rarement 0, à préciser)

Exemple 1 - Voici une série d'instructions (les instructions ci-dessous s'enchaînent **séquentiellement** les une après les autres) appliquées à une liste L. Complétez dans le tableau le contenu progressif de la liste L :

Instructions	Contenu de la liste L après chaque instruction				
	1	2	3	4	5
L = CREER_LISTE_VIDE()					
INSERER(L,28,1)					
INSERER(L,14,2)					
INSERER(L,56,2)					
INSERER(L,37,1)					
INSERER(L,90,2)					

Combien d'éléments contient la liste L après l'exécution de toutes ces instructions ? **Réponse** :

Exemple 2 - Voici une série d'instructions appliquées à une liste. Complétez dans le tableau le contenu de la liste L :

Instructions	Contenu de la liste L après chaque instruction				
	1	2	3	4	5
L = CREER_LISTE_VIDE()					
INSERER(L,'A',1)					
INSERER(L,'O',2)					
INSERER(L,'B',1)					
INSERER(L,'V',3)					
INSERER(L,'R',2)					

Quel est le contenu de la variable L à la fin de ces 6 instructions ? **Réponse** : L = ['.....','.....','.....','.....','.....']

Exemple 3 – Voici une série d'instructions appliquées à une liste. Complétez dans le tableau le contenu de la liste L :

Instructions	Contenu de la liste L après chaque instruction				
	1	2	3	4	5
L = CREER_LISTE_VIDE()					
INSERER(L,23,1)					
INSERER(L,78,1)					
INSERER(L,59,2)					
SUPPRIMER(L,1)					
INSERER(L,63,2)					
MODIFIER(L,97,3)					
INSERER(L,14,2)					
SUPPRIMER(L,3)					
INSERER(L,80,1)					
SUPPRIMER(L,2)					
INSERER(L,44,3)					
INSERER(L,35,1)					
A = LIRE(L,2)					
B = LIRE(L,1)					

Après l'exécution de toutes ces instructions :

- Combien d'éléments contient la liste L ? **Réponse :**
- Combien valent les variables A et B ? **Réponse :** A = B =

Attention : le type « liste » de Python ne correspond pas à la structure de données liste. En python le type « liste » représente un tableau dynamique, qui est un « objet intelligent » possédant de nombreuses méthodes, mais pas directement celles d'une structure de données « liste » décrites dans l'interface ci-dessus.

II - Les piles

On retrouve dans les piles une partie des propriétés vues sur les listes. Dans les piles, il est uniquement possible de manipuler le dernier élément introduit dans la pile. On prend souvent l'analogie avec une pile d'assiettes : dans une pile d'assiettes la seule assiette directement accessible et la dernière assiette qui a été déposée sur la pile.

Les piles sont basées sur le principe LIFO (*Last In First Out* en anglais : le dernier rentré sera le premier à sortir). On retrouve souvent ce principe LIFO en informatique.

Voici les opérations que l'on peut réaliser sur une pile (**interface** d'une pile) :

- CREER_PILE_VIDE() : crée et retourne une pile vide
- EMPILER(P,e) : l'élément e est inséré au sommet de la pile P (**empiler** en français, **push** en anglais)
- DEPILER(P) : l'élément au sommet de la pile P est supprimé et retourné (**dépiler** en français, **pop** en anglais)
- TAILLE(P) : renvoie le nombre d'éléments présents dans la pile P [taille ou longueur de la pile]
- EST_VIDE(P) : retourne Vrai si la pile P est vide [TAILLE(P) = 0]
- EST_PLEINE(P) : retourne Vrai si la pile P est pleine [TAILLE(P) a atteint une valeur maximale]

Exemple d'utilisation d'une pile : soit une pile P composée des éléments suivants : 12, 14, 8, 7, 19 et 22 [le sommet de la pile est 22]. Pour chaque exemple ci-dessous on repart de la pile d'origine :

- DEPILER(P) renvoie 22 et la pile P est maintenant composée des éléments suivants : 12, 14, 8, 7 et 19 [le sommet de la pile est 19]
- EMPILER (P,42) la pile P est maintenant composée des éléments suivants : 12, 14, 8, 7, 19, 22 et 42 [le sommet de la pile est 42]
- si on applique DEPILER (P) 6 fois de suite, EST_VIDE(P) renvoie vrai : la pile est alors vide
- après avoir appliqué DEPILER (P) une fois, TAILLE(P) renvoie 5

Remarque : lorsqu'on représente une pile de manière horizontale (exemple : [7,6,3,9,4]) le **sommet de la pile** est toujours **l'élément de droite** (ici le 4)

Exemple 4 – Voici une série d'instructions appliquées à une pile. Complétez dans le tableau le contenu de la pile P :

Instructions →	P = CREER_PILE_VIDE()	EMPLER(P,17)	EMPLER(P,89)	EMPLER(P,55)	A = DEPILER(P)
Contenu de la pile P →					

Après l'exécution de toutes ces instructions :

- Combien d'éléments contient la pile L ? **Réponse :**
- Combien vaut la variable A ? **Réponse :** A =

Exemple 5 - Voici une série d'instructions appliquées à une pile. Complétez dans le tableau le contenu de la pile P :

Instructions	Contenu de la pile P après chaque instruction					
	→ → → sens de l'empilage (sommet à droite) → → →					
P = CREER_PILE_VIDE()						
EMPILER(P, 17)						
EMPILER(P, 89)						
EMPILER(P, 55)						
A = DEPILER(P)						
EMPILER(P, 41)						
EMPILER(P, 62)						
EMPILER(P, 7)						
EMPILER(P, 93)						
B = DEPILER(P)						
C = DEPILER(P)						
EMPILER(P, 20)						

Après l'exécution de toutes ces instructions :

- Combien valent les variables A, B et C ? **Réponse :** A = B = C =
- Quelle est la valeur placée sur le sommet de la pile P ? **Réponse :** le sommet vaut
- Quelle est la plus ancienne valeur entrée dans la pile P ? **Réponse :**
- Quelle est la valeur entrée dans la pile P le plus récemment ? **Réponse :**
- Quelle est la première valeur entrée dans la pile P ? **Réponse :**
- Quelle est la dernière valeur entrée dans la pile P ? **Réponse :**

Exemple 6 - Voici une série d'instructions appliquées à une pile. Complétez dans le tableau le contenu de la pile P :

Instructions	Contenu de la pile P après chaque instruction					
	→ → → sens de l'empilage (sommet à droite) → → →					
P = CREER_PILE_VIDE()						
EMPILER(P, 8)						
EMPILER(P, 5)						
EMPILER(P, 2)						
A = DEPILER(P)						
EMPILER(P, 4)						
B = DEPILER(P)						
C = DEPILER(P)						

Après l'exécution de toutes ces instructions :

- Combien valent les variables A, B et C ? **Réponse :** A = B = C =
- Quelle est la valeur placée sur le sommet de la pile P ? **Réponse :** le sommet vaut
- Quelle est la plus ancienne valeur entrée dans la pile P ? **Réponse :**

A retenir : le principe d'une pile se résume à « **dernier entré, premier sorti** » (LIFO en anglais). Autrement dit on ne peut accéder qu'à l'élément situé au sommet de la pile [dernier élément entré].

III - Les files

Comme les piles, les files ont des points communs avec les listes. Différences majeures : dans une file on ajoute des éléments à une extrémité de la file et on supprime des éléments à l'autre extrémité. On prend souvent l'analogie de la file d'attente devant un magasin pour décrire une file de données.

Les files sont basées sur le principe FIFO (*First In First Out* en anglais : le premier qui est rentré sera le premier à sortir). Ici aussi, on retrouve souvent ce principe FIFO en informatique.

Voici les opérations que l'on peut réaliser sur une file (**interface** d'une file) :

- CREER_FILE_VIDE() : crée et retourne une file vide
- ENFILER(F,e) : l'élément e est inséré en queue de la file F [**enfiler** en français, **enqueue** en anglais]
- DEFILER(F) : l'élément situé en tête de la file F est supprimé de la file et est retourné [**défiler** en français, **dequeue** en anglais]

- TAILLE(F) : renvoie le nombre d'éléments présents dans la file [taille ou longueur de la file]
- EST_VIDE(F) : retourne Vrai si la file F est vide [TAILLE(F) = 0]
- EST_PLEINE(F) : retourne Vrai si la file F est pleine [TAILLE(F) a atteint une valeur maximale]

Exemple d'utilisation d'une file : soit une file F composée des éléments suivants : 12, 14, 8, 7, 19 et 22 [le premier élément rentré dans la file est 22 ; le dernier élément rentré dans la file est 12]. Pour chaque exemple ci-dessous on repart de la file d'origine :

- ENFILER(F,42) : la file F est maintenant composée des éléments suivants : 42, 12, 14, 8, 7, 19 et 22 [le premier élément rentré dans la file est 22 ; le dernier élément rentré dans la file est 42]
- DEFILER(F) : renvoie le plus ancien élément, soit 22. La file F est maintenant composée des éléments suivants : 12, 14, 8, 7, et 19 [le premier élément rentré dans la file est 19 ; le dernier élément rentré dans la file est 12]
- si on applique DEFILER(F) 6 fois de suite, EST_VIDE(F) renvoie vrai
- après avoir appliqué DEFILER(F) une fois, TAILLE(F) renvoie 5.

Remarque : lorsqu'on représente une file de manière horizontale [exemple : [7,6,3,9,4]] le **premier élément entré** est toujours **l'élément de droite** [ici le 4], et le **dernier élément entré** est **l'élément de gauche** [ici le 7]. L'entrée de la file [appelée « queue » de la file] est alors à **gauche**, et la sortie de la file [appelée « tête » de la file] est à **droite**. Il en résulte que dans une file représentée horizontalement les données circulent **de gauche à droite**.

Exemple 7 - Voici une série d'instructions appliquées à une file. Complétez dans le tableau le contenu de la file F :

Instructions	Contenu de la file F après chaque instruction					
	→ → → sens de la file [queue à gauche et tête à droite] → → →					
F = CREER FILE VIDE()						
ENFILER(F,17)						
ENFILER(F,89)						
ENFILER(F,55)						
A = DEFILER(F)						
ENFILER(F,41)						
ENFILER(F,62)						
ENFILER(F,7)						
ENFILER(F,93)						
B = DEFILER(F)						
C = DEFILER(F)						
ENFILER(F,20)						

Après l'exécution de toutes ces instructions :

- Combien valent les variables A, B et C ? **Réponse** : A = B = C =
- Quelles sont les valeurs en tête et en queue de la file F ? **Réponse** : la tête vaut et la queue vaut
- Quelle est la plus ancienne valeur entrée dans la file F ? **Réponse** :
- Quelle est la valeur entrée dans la file F le plus récemment ? **Réponse** :
- Quelle est la première valeur entrée dans la file F ? **Réponse** :
- Quelle est la dernière valeur entrée dans la file F ? **Réponse** :

Exemple 8 - Voici une série d'instructions appliquées à une file. Complétez dans le tableau le contenu de la file F :

Instructions	Contenu de la file F après chaque instruction					
	→ → → sens de la file [queue à gauche et tête à droite] → → →					
F = CREER FILE VIDE()						
ENFILER(F,3)						
ENFILER(F,7)						
ENFILER(F,1)						
ENFILER(F,4)						
ENFILER(F,8)						
ENFILER(F,5)						
A = DEFILER(F)						
B = DEFILER(F)						

Après l'exécution de toutes ces instructions :

- Combien valent les variables A et B ? **Réponse** : A = B =
- Quelles sont les valeurs en tête et en queue de la file F ? **Réponse** : la tête vaut et la queue vaut

A retenir : le principe d'une file se résume à « **premier entré, premier sorti** » [FIFO en anglais].