

EXERCICES SUR LES LISTES CHAINEES

Exercice1

Définir la structure d'une liste chaînée.

Algorithme :

//structure d'un élément de la liste :

Début

```
struct Cell *next ;  
entier value ;
```

Fin

Cell;

//structure de la liste :

Début

```
Cell *first ;
```

Fin

List ;

Exercice2

Créer une liste chaînée.

Algorithme :

Fonction initialiser() : List*

Début

```
List *list <= allouer(tailleDe(*list));  
Cell *cell <= allouer(tailleDe(*cell)) ;
```

```
cell ->value <= 0 ;  
cell ->next <= NULL ;  
list->first <= cell ;
```

```
Retourner(list) ;
```

Fin

Exercice3

Ajouter un élément en fin de liste chaînée.

Algorithme :

Fonction insererFin(List *list, entier valeur) : vide

Début

Cell *nouv <= allouer(tailleDe(*nouv)) ;

Si (list = NULL) Alors :

retourner();

Fin si

Cell *cell <= list->first ;

Tant que (cell->next != NULL) Faire :

cell <= cell->next ;

Fin tant que

nouv->val <= valeur ;

nouv->next <= NULL ;

cell->next <= nouv ;

Fin

Algorithme d'ajout d'élément en fin de liste :

Début :

LISTE liste ;

CELLULE p ;

Entier i ;

Entier valeur ;

Allouer (p) ;

// Liste vide ?

Si liste = NULL Alors

Afficher(«la liste n'existe pas»);

Retourner 0;

FinSi

// Liste vide ?

Si liste -> tête = NULL Alors

liste -> tête <- p

Sinon

liste -> queue -> suivant <- p ;

FinSi

Liste -> queue <- p ;

p -> val <- valeur ;

p -> suivant <- NULL ;

retourner 1 ;

Fin

Exercice4

Retirer un élément d'un rang quelconque de la de la liste chaînée.

Exercice5

Ajouter un élément à une liste chaînée à un rang quelconque.

Exercice6

Chercher un élément dans une liste chaînée.

“

Exercice7

Supprimer une liste chaînée.

Algorithme :

Fonction delete(List *list) : vide

Début

Tant Que (list→first != NULL) Faire :

Cell *del <= list→first ;

list->first <= list->first->next ;

free(del) ;

FinTantQue

free(liste) ;

Fin

Exercice8

Fusionner 2 listes chaînées déjà triées.

Exercice9

Afficher une liste chaînée.

Algorithme :

Fonction afficher(List *list) : vide

Début

S

Cell *current <= list→first ;

Tant Que (current != NULL) Faire :

Afficher(« %d\n », current->value) ;

current <= current->next ;

FinTantQue

Fin

Exercice10

Incrémenter une liste chaînée binaire.

Exercice11

Décrémenter une liste chaînée binaire