

# E S I

Structures de Données. Deuxième année 'Ingénieur'.  
Correction EMD1. 0809.

SOIT

L , L1 , L2 DES LISTES ;  
Liberer1 , Liberer2 , Liberer3 DES ACTIONS ;  
Afficher UNE ACTION ;  
Nb\_suiv , Nb\_suiv2 : ENTIER ;  
Res UN POINTEUR VERS UNE LISTE ;  
Pluslongueliste UNE ACTION ;  
Compte UN ENTIER ;

DEBUT

Nb\_suiv := 0 ;  
Nb\_suiv2 := 0 ;  
CREER\_LISTE ( L , [ 12 , 34 , 5 , 8 , 45 , 67 , 34 , 10 , 23 , 64 , 87 , 54 , 11 , 43 , 32 , 23 , 17 ] ) ;  
APPEL Liberer1 ( L , L1 ) ;  
Ecrire ( 'Liste résultante après Liberer1' ) ;  
APPEL Afficher ( L1 ) ;  
CREER\_LISTE ( L , [ 12 , 34 , 5 , 8 , 45 , 67 , 34 , 10 , 23 , 64 , 87 , 54 , 11 , 43 , 32 , 23 , 17 ] ) ;  
APPEL Liberer2 ( L , L1 , L2 ) ;  
Ecrire ( 'Liste résultante 1 après Liberer2' ) ;  
APPEL Afficher ( L1 ) ;  
Ecrire ( 'Liste résultante 2 après Liberer2' ) ;  
APPEL Afficher ( L2 ) ;  
APPEL Liberer3 ( L1 , L2 ) ;  
Ecrire ( 'Liste résultante après Liberer3' ) ;  
APPEL Afficher ( L1 ) ;  
Ecrire ( 'nb\_suiv=' , Nb\_suiv , 'nb\_suiv2=' , Nb\_suiv2 ) ;  
CREER\_LISTE ( L , [ 12 , 34 , 5 , 8 , 45 , 67 , 34 , 10 , 23 , 64 , 87 , 54 , 11 , 43 , 32 , 23 , 17 ] ) ;  
APPEL Pluslongueliste ( L , Res , Compte ) ;  
Ecrire ( 'res=' , VALEUR ( Res ) ) ;  
Ecrire ( 'compte=' , Compte ) ;

FIN

{Afficher les éléments d'une liste}

ACTION Afficher ( L )

SOIT

L UNE LISTE ;  
P UN POINTEUR VERS UNE LISTE ;

DEBUT

P := L ;  
TANTQUE P <> NIL  
    Ecrire ( VALEUR ( P ) ) ;  
    P := SUIVANT ( P )  
FINTANTQUE

FIN

{Libérer les éléments de la liste L1 et créer une autre liste L2 ordonnée contenant tous les éléments de L1 }

**ACTION Liberer1 ( L1 , L2 )**

**SOIT**

**L1 , L2 DES LISTES ;**

**P , Q , M , N , Prec DES POINTEURS VERS DES LISTES ;**

**Arret UN BOOLEEN ;**

**DEBUT**

**P := L1 ;**

**L2 := NIL ;**

**TQ P <> NIL**

**Q := SUIVANT ( P ) ;**

**Nb\_suiv := Nb\_suiv + 1 ;**

**ALLOUER ( M ) ;**

**AFF\_VAL ( M , VALEUR ( P ) ) ;**

**AFF\_ADR ( M , NIL ) ;**

**SI L2 = NIL**

**L2 := M**

**SINON**

**{parcourir la liste L2 }**

**N := L2 ;**

**Arret := FAUX ;**

**Prec := NIL ;**

**TQ NON Arret ET ( N <> NIL )**

**SI VALEUR ( P ) >= VALEUR ( N )**

**Prec := N ;**

**N := SUIVANT ( N ) ;**

**Nb\_suiv := Nb\_suiv + 1 ;**

**SINON**

**Arret := VRAI**

**FSI ;**

**FTQ ;**

**SI Arret**

**SI Prec <> NIL**

**AFF\_ADR ( Prec , M ) ;**

**AFF\_ADR ( M , N )**

**SINON**

**AFF\_ADR ( M , N ) ;**

**L2 := M**

**FSI**

**SINON**

**AFF\_ADR ( Prec , M )**

**FSI**

**FSI ;**

**LIBERER ( P ) ;**

**P := Q ;**

**FTQ**

**FIN**

{Libérer les éléments de la liste L1 et créer deux autres listes L1 et L2. L1 contient tous les de L1 qui sont déjà dans l'ordre L2 contient les autres }

**ACTION Liberer2 ( L , L1 , L2 )**

**SOIT**

**L , L1 , L2 DES LISTES ;**

**P , Q , M , Der1 , Der2 DES POINTEURS VERS DES LISTES ;**

**DEBUT**

**P := L ;**

**L1 := NIL ;**

**L2 := NIL ;**

**TQ P <> NIL**

**Q := SUIVANT ( P ) ;**

**Nb\_suiv2 := Nb\_suiv2 + 1 ;**

**ALLOUER ( M ) ;**

**AFF\_VAL ( M , VALEUR ( P ) ) ;**

**AFF\_ADR ( M , NIL ) ;**

**SI L1 = NIL**

**L1 := M ;**

**Der1 := M**

**SINON**

**SI VALEUR ( P ) >= VALEUR ( Der1 )**

**AFF\_ADR ( Der1 , M ) ;**

**Der1 := M**

**SINON**

**SI L2 = NIL**

**L2 := M ;**

**Der2 := M**

**SINON**

**AFF\_ADR ( Der2 , M ) ;**

**Der2 := M**

**FSI**

**FSI**

**FSI ;**

**LIBERER ( P ) ;**

**P := Q ;**

**FTQ**

**FIN**

{Libérer les éléments de la liste L2 pour les rajouter aux éléments de L1 tout en préservant l'ordre croissant }

**ACTION Liberer3 ( L1 , L2 )**

**SOIT**

**L1 , L2 DES LISTES ;**

**P , Q , M , N , Prec DES POINTEURS VERS DES LISTES ;**

**Arret UN BOOLEEN ;**

**DEBUT**

**P := L2 ;**

**TQ P <> NIL**

**Q := SUIVANT ( P ) ;**

**Nb\_suiv2 := Nb\_suiv2 + 1 ;**

**ALLOUER ( M ) ;**

**AFF\_VAL ( M , VALEUR ( P ) ) ;**

**AFF\_ADR ( M , NIL ) ;**

**{parcourir la liste L1 }**

**N := L1 ;**

**Arret := FAUX ;**

**Prec := NIL ;**

**TQ NON Arret ET ( N <> NIL )**

**SI VALEUR ( P ) >= VALEUR ( N )**

**Prec := N ;**

**N := SUIVANT ( N ) ;**

**Nb\_suiv2 := Nb\_suiv2 + 1 ;**

**SINON**

**Arret := VRAI**

**FSI ;**

**FTQ ;**

**SI Arret**

**SI Prec <> NIL**

**AFF\_ADR ( Prec , M ) ;**

**AFF\_ADR ( M , N )**

**SINON**

**AFF\_ADR ( M , N ) ;**

**L1 := M**

**FSI**

**SINON**

**AFF\_ADR ( Prec , M )**

**FSI ;**

**LIBERER ( P ) ;**

**P := Q ;**

**FTQ**

**FIN**

{Plus longue liste }

**ACTION Pluslongueliste ( L , Result , Longmax )**

**SOIT**

**L , Res , P , Q , Result DES POINTEURS VERS DES LISTES ;**

**Compte , Minl , Mincour , Longmax DES ENTIERS ;**

**DEBUT**

**Minl := + 1000 ;**

**Longmax := 0 ;**

**P := L ;**

**TQ P <> NIL**

**SI VALEUR ( P ) >= Minl**

**Minl := VALEUR ( P ) ;**

**SINON**

**Compte := 1 ;**

**Minl := VALEUR ( P ) ;**

**Mincour := Minl ;**

**Res := P ;**

**ECRIRE ( '==>' , Mincour , Compte ) ;**

**Q := SUIVANT ( P ) ;**

**TQ Q <> NIL**

**SI VALEUR ( Q ) >= Mincour**

**Mincour := VALEUR ( Q ) ;**

**Compte := Compte + 1 ;**

**ECRIRE ( '==>' , Mincour , Compte ) ;**

**FSI ;**

**Q := SUIVANT ( Q )**

**FTQ ;**

**ECRIRE ( 'RES=' , VALEUR ( Res ) ) ;**

**ECRIRE ( 'COMPTE=' , Compte ) ;**

**SI Compte > Longmax**

**Longmax := Compte ;**

**Result := Res**

**FSI**

**FSI ;**

**P := SUIVANT ( P )**

**FTQ**

**FIN**

## Résultats du programme

Liste résultante après Liberer1

5  
8  
10  
11  
12  
17  
23  
23  
32  
34  
34  
43  
45  
54  
64  
67  
87

Liste résultante 1 après Liberer2

12  
34  
45  
67  
87

Liste résultante 2 après Liberer2

5  
8  
34  
10  
23  
64  
54  
11  
43  
32  
23  
17

Liste résultante après Liberer3

5  
8  
10  
11  
12  
17  
23  
23  
32  
34  
34  
43  
45  
54  
64  
67  
87

nb\_suiv= 92    nb\_suiv2= 84

==> 12 1

==> 34 2

==> 45 3

==> 67 4

==> 87 5

RES= 12

COMPTE= 5

==> 5 1

==> 8 2

```
==> 45 3
==> 67 4
==> 87 5
RES= 5
COMPTE= 5
==> 34 1
==> 64 2
==> 87 3
RES= 34
COMPTE= 3
==> 10 1
==> 23 2
==> 64 3
==> 87 4
RES= 10
COMPTE= 4
==> 54 1
RES= 54
COMPTE= 1
==> 11 1
==> 43 2
RES= 11
COMPTE= 2
==> 32 1
RES= 32
COMPTE= 1
==> 23 1
RES= 23
COMPTE= 1
==> 17 1
RES= 17
COMPTE= 1
res= 12
compte= 5
```