EMP
Structures de données
Examen1 // 2011 2012
Barème probable : (6 + 6 + 4 + 2)
Durée : 2H.

Liste linéaire chaînée de tableaux

1. On veut construire une liste de tableaux de taille (N+1) à partir de X données aléatoires de la manière suivante :
Les N premières données sont rangées dans le tableau du premier maillon,
Les N suivantes dans le tableau du deuxième maillon,
Etc.

Le premier élément du tableau contient le nombre d'éléments dans le tableau.

Les tableaux peuvent être statiques ou dynamiques.

X >> N.

Utiliser la fonction Rand(Max) qui génère un nombre aléatoire entre 0 et Max-1. Max >> X.

2. On veut afficher tous les éléments de la liste créée en 1. en ordre croissant de la manière suivante :
Répéter jusqu'à ce que la liste devienne vide :
- Chercher le plus petit élément, soit e, et l'afficher.
- Enlever e du tableau correspondant. Si ce dernier devient vide, le maillon le contenant est libéré.

Utiliser le module Min_liste (L, Prec, P, i) donnant le maillon P de la liste L contenant le plus petit élément de la liste L et sa position i dans le tableau correspondant. Prec est le maillon qui précède P.

3. Ecrire le module Min_liste(L, Prec, P, i). Ecrire préalablement le module Min_Tab(T, Min, Ind_min) qui recherche le plus petit élément Min d'indice Ind_min dans le tableau T.

4. Programmer l'algorithme 1. en C.

EMP
Structures de données
Corrigé Examen1 // 2011 2012

Liste linéaire chaînée de tableaux

```
SOIT
   L UNE LISTE DE TABLEAUX ( 11 ) DYNAMIQUES ;
   X UN ENTIER ;
   Creerliste , Afficherliste , Afficherliste2 DES ACTIONS ;
   Sup UNE ACTION ;
   Min_tab , Min_liste DES ACTIONS ;
   Taille : ENTIER ;
   Ind_mini : ENTIER ;

DEBUT
   Taille := 11 ;
   LIRE ( X ) ;
   APPEL Creerliste ( L , X ) ;
   APPEL Afficherliste ( L ) ;
   ECRIRE ( 'Affichage en ordre croissant' ) ;
   APPEL Afficherliste2 ( L , X ) ;

FIN
ACTION Min_tab ( T , Plus_petit , Ind_min )
SOIT
   T UN TABLEAU ( 11 ) DYNAMIQUE ;
   Plus_petit , Ind_min : ENTIERS ;
   I UN ENTIER ;

DEBUT
   Plus_petit := ELEMENT ( T [ 2 ] ) ;
   Ind_min := 2 ;
   POUR I := 3 , ELEMENT ( T [ 1 ] ) + 1
      SI ELEMENT ( T [ I ] ) < Plus_petit
         Plus_petit := ELEMENT ( T [ I ] ) ;
         Ind_min := I
      FSI
   FPOUR
FIN

ACTION Min_liste ( Liste_ , Q_res , P_res , Ind_res )
SOIT
   Liste_ , L , P_res , Q_res , P_res2 DES LISTES DE TABLEAUX ( 11 ) DYNAMIQUES ;
   Ind_res , Ind_res2 DES ENTIERS ;
   V_res , V_res2 DES ENTIERS ;

DEBUT
   Q_res := NIL ;
   L := Liste_ ;
   APPEL Min_tab ( VALEUR ( L ) , V_res , Ind_res ) ;
   P_res := L ;
   L := SUIVANT ( L ) ;
   TQ L <> NIL
      APPEL Min_tab ( VALEUR ( L ) , V_res2 , Ind_res2 ) ;
      SI V_res2 < V_res
         V_res := V_res2 ;
         P_res := L ;
         Ind_res := Ind_res2
```

```
    FSI ;
    Q_res := L ;
    L := SUIVANT ( L )
  FTQ
FIN
ACTION Creerliste ( L , N )
SOIT
  L UNE LISTE DE TABLEAUX ( 11 ) DYNAMIQUES ;
  N , I , M DES ENTIERS ;
  T UN TABLEAU ( 11 ) DYNAMIQUE ;
  K UN ENTIER ;
  P , Q DES LISTES DE TABLEAUX ( 11 ) DYNAMIQUES ;

DEBUT
  K := 0 ;
  L := NIL ;
  ALLOC_TAB ( T ) ;
  POUR I := 1 , N
    K := K + 1 ;
    SI ( K <= Taille - 1 )
      AFF_ELEMENT ( T [ K + 1 ] , ALEANOMBRE ( 1000 ) )
    FSI ;
    SI ( K = Taille - 1 ) OU ( I = N )
      AFF_ELEMENT ( T [ 1 ] , K ) ;
      ALLOUER ( Q ) ;
      AFF_VAL ( Q , T ) ;
      SI L = NIL
        L := Q
      SINON
        AFF_ADR ( P , Q ) ;

      FSI ;
      P := Q ;
      ALLOC_TAB ( T ) ;
      K := 0 ;

    FSI
  FPOUR ;
  AFF_ADR ( Q , NIL ) ;

FIN
ACTION Afficherliste ( L )
SOIT
  L UNE LISTE DE TABLEAUX ( 11 ) DYNAMIQUE ;

DEBUT
  SI L <> NIL
    ECRIRE ( 'Tableau:' ) ;
    ECRIRE ( VALEUR ( L ) ) ;
    APPEL Afficherliste ( SUIVANT ( L ) )
  FSI
FIN
ACTION Afficherliste2 ( L , N )
SOIT
  L , Mini , Prec DES LISTES DE TABLEAUX ( 11 ) DYNAMIQUE ;
  N UN ENTIER ;
  Ind_min UN ENTIER ;

DEBUT
  TQ L <> NIL
```

```
      APPEL Min_liste ( L , Prec , Mini , Ind_min ) ;
      ECRIRE ( ELEMENT ( VALEUR ( Mini ) [ Ind_min ] ) ) ;
      APPEL Sup ( L , Prec , Mini , Ind_min ) ;

   FTQ
FIN
ACTION Sup ( L , Prec , P , Ind )
SOIT
   L , P , Prec , Mini DES LISTE DE TABLEAUX ( 11 ) DYNAMIQUES ;
   Ind , K : ENTIER ;

DEBUT
{supprimer l'élément du tableau }
   SI Ind <> ELEMENT ( VALEUR ( P ) [ 1 ] ) + 1
      K := ELEMENT ( VALEUR ( P ) [ 1 ] ) ;
      AFF_ELEMENT ( VALEUR ( P ) [ Ind ] , ELEMENT ( VALEUR ( P ) [ K + 1 ] ) )
   FSI ;
   AFF_ELEMENT ( VALEUR ( P ) [ 1 ] , ELEMENT ( VALEUR ( P ) [ 1 ] ) - 1 ) ;
  { Si tableau vide , liberer le maillon }
   SI ELEMENT ( VALEUR ( P ) [ 1 ] ) = 0
      SI Prec = NIL
         L := SUIVANT ( L )
      SINON
         AFF_ADR ( Prec , SUIVANT ( P ) )
      FSI
   FSI
FIN
```