```
SOIT
   A UN ARB ;
   Premier_inordre , Premier_postordre DES FONCTIONS ( ARB ) ;
   Liberer_postordre , Liberer_inordre , Liberer_preordre DES ACTIONS ;

DEBUT
   CREER_ARB ( A , [ 55 , 60 , 99 , 45 , 32 , 43 , 44 , 38 , 40 , 50 , 52 , 48 , 41 ] ) ;
   ECRIRE ( 'Premier post=' , INFO ( Premier_postordre ( A ) ) ) ;
   ECRIRE ( 'post' ) ;
   APPEL Liberer_postordre ( A ) ;
   CREER_ARB ( A , [ 55 , 60 , 99 , 45 , 32 , 43 , 44 , 38 , 40 , 50 , 52 , 48 , 41 ] ) ;
   ECRIRE ( 'Premier in=' , INFO ( Premier_inordre ( A ) ) ) ;
   ECRIRE ( 'in' ) ;
   APPEL Liberer_inordre ( A ) ;
   CREER_ARB ( A , [ 55 , 60 , 99 , 45 , 32 , 43 , 44 , 38 , 40 , 50 , 52 , 48 , 41 ] ) ;
   ECRIRE ( 'Pr' ) ;
   APPEL Liberer_preordre ( A ) ;
FIN
FONCTION Premier_inordre ( Arbre ) : ARB ;
SOIT
   Arbre UN ARB ;
   P UN ARB ;

DEBUT
   P := Arbre ;
   TQ FG ( P ) <> NIL
      P := FG ( P )
   FTQ ;
   Premier_inordre := P
FIN
FONCTION Premier_postordre ( Arbre ) : ARB ;
SOIT
   Arbre UN ARB ;
   Continue UN BOOLEEN ;
   P UN ARB ;

DEBUT
   P := Arbre ;
   Continue := VRAI ;
   TQ Continue
      TQ FG ( P ) <> NIL
         P := FG ( P )
      FTQ ;
      SI FD ( P ) <> NIL
         P := FD ( P )
      SINON
         Continue := FAUX ;
         Premier_postordre := P
      FSI
   FTQ
FIN
```

```
ACTION Liberer_postordre ( Arbre ) ;
SOIT
   Arbre UN ARB ;
   Continue UN BOOLEEN ;
   P , Parent , Sauv_p DES ARB ;
DEBUT
   P := Premier_postordre ( Arbre ) ;
   TQ P <> NIL
      Parent := PERE ( P ) ;
      Sauv_p := P ;
      ECRIRE ( INFO ( P ) ) ;
      LIBERERNOEUD ( P ) ;
      SI Parent <> NIL
         SI FD ( Parent ) = Sauv_p
            P := Parent ;
         SINON
            SI FD ( Parent ) = NIL
               P := Parent ;
            SINON
               P := Premier_postordre ( FD ( Parent ) ) ;
            FSI
         FSI
      SINON
         P := NIL
      FSI
   FTQ
FIN

ACTION Liberer_inordre ( Arbre ) ;
SOIT
   Arbre UN ARB ;
   P , Parent , Sauv_p , Sauv_fd_p DES ARB ;
DEBUT
   P := Premier_inordre ( Arbre ) ;
   TQ P <> NIL
      Parent := PERE ( P ) ;
      Sauv_p := P ;
      Sauv_fd_p := FD ( P ) ;
      ECRIRE ( INFO ( P ) ) ;
      LIBERERNOEUD ( P ) ;
      SI Parent <> NIL
         AFF_FG ( Parent , Sauv_fd_p ) ;
      FSI ;
      SI Sauv_fd_p <> NIL
         AFF_PERE ( Sauv_fd_p , Parent ) ;
         P := Premier_inordre ( Sauv_fd_p ) ;
      SINON
         P := Parent ;
      FSI
   FTQ
FIN
```

```
ACTION Liberer_preordre ( Arbre ) ;
SOIT
   Arbre UN ARB ;        P , Q DES ARB ;
DEBUT
   P := Arbre ;
   TQ P <> NIL
      SI ( FG ( P ) <> NIL ) ET ( FD ( P ) <> NIL )
         ECRIRE ( 'Sup' , INFO ( P ) ) ;
         Q := FG ( P ) ;
         TQ (FD ( Q ) <> NIL)  OU (FG ( Q ) <> NIL)
            SI FD(Q) <> NIL  Q := FD ( Q ) SINON Q := FG(Q) FSI
         FTQ ;
         AFF_INFO ( P , INFO ( Q ) ) ;
         SI  Q = FD( PERE( Q ) )
            AFF_FD ( PERE ( Q ) , NIL );
         SINON
            AFF_FG ( PERE ( Q ) , NIL )
         FSI ;
         SI P <> PERE ( Q )    P := FG ( P )    FSI;
         LIBERERNOEUD ( Q ) ;
      SINON
         SI ( FG ( P ) <> NIL ) ET ( FD ( P ) = NIL )
            SI PERE ( P ) <> NIL
               AFF_FG ( PERE ( P ) , FG ( P ) ) ;
            FSI ;
            AFF_PERE ( FG ( P ) , PERE ( P ) ) ;
            ECRIRE ( 'Lib' , INFO ( P ) ) ;
            Q := P ;    P := FG ( P ) ;    LIBERERNOEUD ( Q ) ;
         SINON
            SI ( FG ( P ) = NIL ) ET ( FD ( P ) <> NIL )
               SI PERE ( P ) <> NIL
                  AFF_FG ( PERE ( P ) , FD ( P ) ) ;
               FSI ;
               AFF_PERE ( FD ( P ) , PERE ( P ) ) ;
               ECRIRE ( 'Lib' , INFO ( P ) ) ;
               Q := P ;     P := FD ( P ) ;   LIBERERNOEUD ( Q ) ;
            SINON
               ECRIRE ( 'Lib' , INFO ( P ) ) ;
               SI PERE ( P ) <> NIL
                  AFF_FG ( PERE ( P ) , NIL )
               SINON
                  Arbre := NIL
               FSI ;
               Q := P ;
               P := PERE ( P ) ;
               LIBERERNOEUD ( Q ) ;
            FSI
         FSI
      FSI
   FTQ
FIN
```

```
ACTION Liberer_preordre ( A ) ; // Solution 2
SOIT
   A , B DES ARB ;
   Stop UN BOOLEEN ;
DEBUT
   Stop := FAUX ;
   TQ NON Stop
      SI A <> NIL
         ECRIRE ( INFO ( A ) ) ;
         B := A ;
         SI FG ( A ) <> NIL
            SI FD ( A ) <> NIL
               AFF_PERE ( FD ( A ) , PERE ( A ) ) ;
               AFF_PERE ( FG ( A ) , FD ( A ) ) ;
            SINON
               AFF_PERE ( FG ( A ) , PERE ( A ) )
            FSI ;
            A := FG ( A )
         SINON
            SI FD ( A ) <> NIL
               AFF_PERE ( FD ( A ) , PERE ( A ) ) ;
               A := FD ( A ) ;
            SINON
               SI PERE ( A ) <> NIL
                  A := PERE ( A )
               SINON
                  Stop := VRAI
               FSI
            FSI
         FSI
      FSI ;
      LIBERERNOEUD ( B )
   FTQ ;
FIN
```