

```

/*****
/* Recherche avec Best First Search ( profondeur DFS) */
/* Suppose que chaque noeud contient l'estimation de la */
/* restante */
/* Utilise une liste de noeuds */
/* Prendre ... chaque tour le plus petit */
*****/
SOIT
    G UN GRAPHE DE ENTIERS ;
    I , Indmin DES ENTIERS ;
    N , N1 , N2 , N3 , N4 , N5 , N6, n7, n8 DES NOEUDS ( G ) ;
    List UNE LISTE DE NOEUD ( G ) ;
    Trouv UN BOOLEEN ;
    Pluspetit : ACTION ;

DEBUT
    { Un arbre m-aire }
    CREER_GRAPHE ( G , [ N1 ( 5 ) , N2 ( 8 ) , N3 ( 6 ) , N4 ( 3 ) , N5 ( 7 ) , N6 ( 8 ) ,
n7(11), n8(4) ]
    UNION [ N1 -> N2 , N1 -> N3 , N1 -> N4 , N3 -> N5 , N3 -> N6, n2->n7, n6->n8 ] ) ;
    CREERLISTE ( List ) ;
    INSERER ( List , N1 , 1 ) ;
    Trouv := FAUX ;
    TQ ( NBRLISTE ( List ) > 0 ) ET ( NON Trouv )
        APPEL Pluspetit ( List , Indmin , N ) ;
        ECRIRE ( 'Noeud : ' , VALEUR ( N ) ) ;
        SI VALEUR ( N ) = 43
            Trouv := VRAI
        SINON
            SUPPRIMER ( List , Indmin ) ;
            SI DEGRE ( N ) <> 0
                POUR I := 1 , DEGRE ( N )
                    INSERER ( List , ADJACENT ( N , I ) , NBRLISTE ( List ) + 1 )
                FINPOUR
            FSI
        FSI
    FTQ ;
    SI Trouv
        ECRIRE ( 'S U C C E S' )
    SINON
        ECRIRE ( 'E C H E C' )
    FSI
FIN
ACTION Pluspetit ( L , Indice_min , N )
SOIT
    Indice_min : ENTIER ;
    L UNE LISTE DE NOEUD ( G ) ;
    M , N DES NOEUD ( G ) ;

```

```
DEBUT
  N_IEME ( L , 1, n ) ;
  I := 2 ;
  Indice_min := 1 ;
  TQ I <= NBRLISTE ( L )
    SUIVANT ( L, m ) ;
    SI VALEUR ( M ) < VALEUR ( N )
      Indice_min := I ;
      N := M ;
    FSI ;
    I := I + 1
  FTQ ;

FIN
```