Feuille de Travaux Dirigés 1

Exercice 1 En utilisant les règles de portée de Pascal, déterminer pour chaque occurrence des noms a et b dans le programme suivant, quelle déclaration s'y applique. Que fait ce programme ?

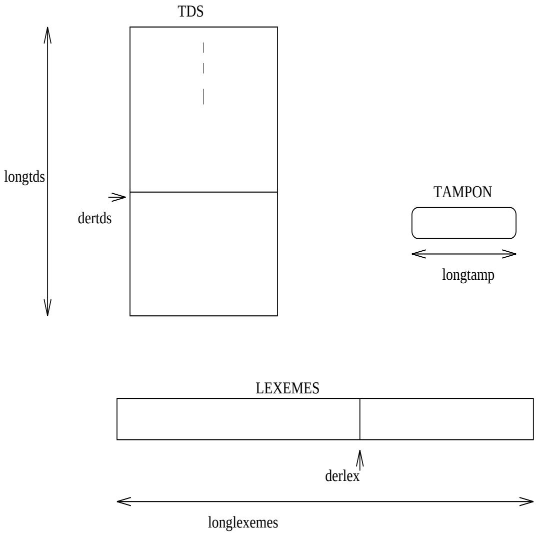
```
program a;
  procedure b(u, v, x, y : integer);
     var a : record a, b : integer end;
         b : record b, a : integer end;
     begin with a do begin a := u; b := v end;
           with b do begin a := x; b := y end;
           writeln(a.a, a.b, b.a, b.b)
     end;
begin
  b(1, 2, 3, 4)
end.
```

Exercice 2 Pour la configuration de la table des symboles vue en cours, et pour le programme Pascal suivant, indiquez l'état de la tds lorsque le reconnaisseur d'identificateurs est situé au niveau de *.

```
program param;
  procedure b(function h (n:integer):integer);
    var m : integer;
    begin m := 3; writeln(h(2)) end {b} ;

  procedure c;
    var m : integer;

    function f (n : integer) : integer;
      begin f := m + n end {f} ;

    procedure r ;
      var m : integer;
      begin {*} m := 7; b(f) end {r} ;

  begin m := 0; r end {c} ;

begin
  c
end.
```

Quelle est la valeur affichée par le programme ?

Exercice 3 Ecrire la fonction d'insertion d'un nouvel identificateur dans la table des symboles dont la structure est la suivante :



- *dertds* est la prochaine place libre dans la tds.
- *derlex* est la prochaine place libre dans le vecteur Lexemes.
- *longtamp* est la longueur du tampon.
- *longlexemes* est la taille du vecteur Lexemes.
- *longtds* est la taille de la tds.
- *longsymb* est la longueur du lexème reconnu.

Cette fonction retourne l'adresse de l'identificateur inséré.

Pour cette même structure de la tds, écrire la fonction qui recherche un identificateur dans la tds. Cette fonction retournera 0 si l'identificateur ne figure pas dans la tds.

Exercice 4 Inférer le type polymorphe de la fonction *longueur* dont le corps d'instructions est le suivant :

```
fun longueur(pliste) =
        if null(pliste) then 0
        else longueur( tl(pliste)) + 1 ;
```

Même question pour la procédure *append* :

```
fun append(l1, l2) =
        if null(l1) then l2
        else cons(hd(l1), append(tl(l1), l2)) ;
```

Même question pour la procédure *mappe* :

```
fun mappe (f, l) =
        if null(l) then nil
        else cons( f( hd(l)), mappe(f, tl(l)) );
```