Université d'Orléans **Maîtrise d'Informatique**

Feuille de Travaux Dirigés 7

Exercice 1 Pour le fragment de programme ci-après :

1. Indiquer l'arbre d'imbrication pour le programme p.
2. Donner deux arbres de récursion (ou arbres des appels) différents pour le programme p.
3. Donner les profondeurs d'imbrication de toutes les procédures du programme correspondant à leur définition et à leur utilisation.
4. Ecrire un algorithme qui construit l'arbre des prédécesseurs statiques à partir de l'arbre de récursion et de l'arbre d'imbrication. Appliquer cet algorithme aux arbres donnés en 1. et 2. pour produire les arbres des prédécesseurs statiques correspondants.

```
program p;
  procedure q;
    procedure t;
       procedure v;
       begin
       [
       end;
       procedure u;
       begin
       [
       end;
    begin
    u
    end;
  begin t;
        if b' then q fi;
  end;
  procedure r;
  begin if b then q else s fi;
  end;
  procedure s ;
  begin
  [
  end;
begin s;
      q;
      r;
      s;
end.
```

Exercice 2 Traduire en code SPARC virtuel le programme suivant :

```
program p;
  var a, b, i :integer;
  procedure perm(var i, j: integer);
    var temp : integer;
    begin temp := i;
          i := j;
          j := temp
    end;
  begin a:=2;
        i := 1;
        if i <> 3 then begin b := 3;
                             perm(a, b)
                       end;
  end.
```

Exercice 3 Même question pour le programme suivant :

```
program main;
type t = record a : array[-5..+5, 1..9] of integer;
                b : ^t
         end;
var i, j : integer;
    box : t;
    pt : ^t;

function lier_copie(var t1: t) : ^t;
var i, j : integer;
begin new(t1.b);
      for i := -5 to 5 do
        for j := 1 to 9 do
           t1^.b^.a[i,j] := t1.a[0, 1];
      t1.b^.b := nil;
      lier_copie := t1.b
end;

begin i := 5;
      repeat j := 1;
             repeat box.a[i, j] := 0;
                    j := j+1
             until j > 9;
             i := i + 1
      until i > 5;
      box.b := nil;
      pt := lier_copie(box)
end.
```