

Práctico 8 - Alcance y Pasaje de Parámetros

Programación 1
InCo - Facultad de Ingeniería, Udelar

1. Dado el siguiente programa:

```
program Ejercicio1 (input, output);
var a, b : integer;

procedure a1 (var x : Real);
var c : char;

    procedure a2 (y : char);
    var d : integer;
    begin
        ... {sentencias de a2}
    end;

begin
    ... {sentencias de a1}
end;

procedure b1;
var e : integer;
begin
    ... {sentencias de b1}
end;

begin
    ... {sentencias de principal}
end.
```

(a) Indique las variables a las que se puede hacer referencia dentro del alcance del procedimiento a2:

a b c d e

(b) Indique las variables a las que se puede hacer referencia dentro del alcance del procedimiento a1:

a b c d e

(c) Indique las variables a las que se puede hacer referencia dentro del alcance del procedimiento b1:

a b c d e

(d) Indique a cuáles procedimientos podría llamar (invocar) el programa principal:

a1 a2 b1

La función a1, aunque está en el alcance del programa principal, no se podría invocar dado que tiene un parámetro por referencia de tipo real y no hay variables de tipo real definidas en el programa principal.

2. Determine la salida del siguiente programa cuando se ingresa en la entrada el valor 4.

```
program Ejercicio2 (input, output);
var x : integer;

procedure procA (var y : integer);
begin
    y := y * 2;
end;
```

```

end;

procedure procB (x : integer);
begin
  x := x + 5;
end;

begin
  readln (x);
  procB (x);
  writeln (x);
  procA (x);
  writeln (x)
end.

```

La salida exhibida es la siguiente:

```

4
8

```

3. Dado el siguiente programa

```

program Ejercicio3 (output);
var num, aux, temp : integer;

procedure pascal (num : integer; var suma : integer);
var aux : integer;
begin
  temp := 2 * num + suma;
  aux := temp * suma;
  num := num + 1;
  suma := suma + num;
  writeln (temp, aux, num, suma)
end;

begin
  num := 1;
  aux := 0;
  temp := 0;
  pascal (num, temp);
  writeln (num, aux, temp);
  temp := temp + 1;
  pascal (temp, num);
  writeln (num, aux, temp)
end.

```

(a) Determine la salida que exhibirá el programa.

```

4 4 2 4
1 0 4
11 11 6 7
7 0 11

```

(b) Indique cuáles de las siguientes afirmaciones son verdaderas:

- El alcance de la variable `temp` no incluye al procedimiento.
- El programa principal no puede hacer referencia a la variable local `aux` del procedimiento.**
- El procedimiento se podría colocar después del programa principal (luego del punto final).
- El parámetro variable `suma` debe ir antes del parámetro por valor `num`.

(c) Indique cuáles de las siguientes afirmaciones son falsas:

- El procedimiento puede hacer referencia a la variable global `temp`.
- El alcance de la variable local `aux` es el procedimiento.
- El primer `writeln` en el programa principal no exhibirá el contenido de la variable local `aux` del procedimiento.

- El parámetro `num` en la declaración del procedimiento es un parámetro *formal*.
- El parámetro `num` en la instrucción que llama al procedimiento es un parámetro *verdadero*.
- El parámetro formal `num` del procedimiento es un identificador local.
- El alcance de la variable global `num` no incluye al procedimiento.

La última opción no es falsa debido a que si bien el alcance de la variable `num` incluiría al procedimiento, en este caso no lo hace dado que hay un identificador local con el mismo nombre.

4. Determine la salida que exhibirá el siguiente programa:

```

program Final (input, output);
var manzanas, bananas, naranjas : integer;

procedure grado (a, b : integer; var c : integer);
var manzanas : integer;

    procedure que (var d : integer);
    begin
        d := d * 2
    end;

begin
    que (b);
    manzanas := 16;
    que (c)
end;

begin
    manzanas := 2;
    bananas := 3;
    naranjas := 6;
    grado (manzanas, bananas, naranjas);
    writeln (manzanas, bananas, naranjas)
end.

```

La salida exhibida es la siguiente:

2 3 12

5. Para cada uno de los siguientes esqueletos de programa, indique qué identificadores son accesibles (es decir: es posible utilizar) dentro de cada bloque (procedimiento `y`, procedimiento `z` o programa `x`), lo que representan (constantes o variables) y dónde fueron declarados (procedimiento `y`, procedimiento `z` o programa `x`)

(a)

```

program x;
const
    a = ...;
    b = ...;
    c = ...;
var
    d : ...;
    e : ...;
    f : ...;

procedure y;
var
    b : ...;
    e : ...;

procedure z;
var
    f : ...;
    g : ...;
begin
    ... { Sentencias del procedimiento z }
end;

begin
    ... { Sentencias del procedimiento y }
end.

```

```

end;

begin
  ... { Sentencias del programa x }
end.

```

Solución:

- Las constantes **a** y **c**, declaradas en el programa **x**, son accesibles desde el programa **x**, el procedimiento **y** y el procedimiento **z**.
- La constante **b**, declarada en el programa **x**, es accesible únicamente desde el programa **x**.
- La variable **d**, declarada en el programa **x**, es accesible desde el procedimiento **y**, el procedimiento **z** y desde el programa **x**.
- La variable **e**, declarada en el programa **x**, es accesible únicamente desde el programa **x**.
- La variable **f**, declarada en el programa **x**, es accesible dentro del procedimiento **y** y desde el programa **x**.
- Las variables **b** y **e**, declaradas en el procedimiento **y**, son accesibles desde el procedimiento **y** y el procedimiento **z**.
- Las variables **f** y **g**, declaradas en el procedimiento **z**, son accesibles desde el procedimiento **z**.

(b)

```

program x;
const
  a = ...;
  b = ...;
  c = ...;
var
  d : ...;
  e : ...;
  f : ...;

procedure y;
var
  b : ...;
  c : ...;

  procedure z;
  var
    f : ...;
    g : ...;
  begin
    ... { Sentencias del procedimiento z }
  end;

begin
  ... { Sentencias del procedimiento y }
end;

begin
  ... { Sentencias del programa x }
end.

```

Solución:

- La constante **a**, declarada en el programa **x**, es accesible desde el procedimiento **y**, el procedimiento **z** y el programa **x**.
- Las constantes **b** y **c**, declaradas en el programa **x**, son accesibles únicamente dentro del programa **x**.
- Las variables **d** y **e**, declaradas en el programa **x**, son accesibles dentro del programa **x** y desde los procedimiento **y** y **z**.

- La variable `f`, declarada dentro del programa `x`, es accesible dentro del programa `x` y del procedimiento `y`.
- Las variables `b` y `c` declaradas dentro del procedimiento `y`, son accesibles dentro del procedimiento `y` y dentro del procedimiento `z`.
- Las variables `f` y `g` declaradas dentro del procedimiento `z`, son accesibles únicamente dentro del procedimiento `z`.

6. (parcial año 2000)

Indique cuál será la salida del siguiente programa si se ingresa el último dígito de su número de cédula de identidad.

```

program parcial2000 (input,output);
var z: integer;

function f (y : integer) : boolean;
begin
  z := z + y;
  f := y mod 2 = 0
end;

procedure a (var y : integer);
begin
  y := 2 * y
end;

procedure b (var z : integer);
begin
  z := z + 5
end;

begin
  readln (z);
  if f(z) then a(z) else b(z);
  writeln ('Salida = ', z)
end.

```

0	1	2	3	4	5	6	7	8	9
0	7	8	11	16	15	24	19	32	23

7. (examen febrero 2019)

Indique cuál será la salida del siguiente programa si se le ingresa el último dígito de su número de cédula de identidad.

```

program estival;
var x,y,z: integer;

procedure atlantida (var x: integer; y: integer);
begin
  y := x - 10;
  x := 5;
  z := x + y + z;
  writeln (z)
end;

function lapaloma (a: integer; b: integer) : integer;
var x,z: integer;

  procedure ptadeleste (var x: integer);
  begin
    x := x + z
  end;

```

```
begin
  z := a;
  x := b;
  ptadeleste (y);
  y := x - z;
  lapaloma := y + x
end;
begin
  readln (x);
  y := x + 1;
  z := x - 1;
  atlantida (y, z);
  z := lapaloma (x, y);
  writeln (x);
  writeln (y);
  writeln (z)
end.
```

0	1	2	3	4	5	6	7	8	9
-5	-3	-1	1	3	5	7	9	11	13
0	1	2	3	4	5	6	7	8	9
5	4	3	2	1	0	-1	-2	-3	-4
10	9	8	7	6	5	4	3	2	1