# Recursividad

En cuanto al uso de subprogramas, existen las siguientes restricciones:

- Todo subprograma que sea invocado por otro subprograma debe ser declarado previamente dentro del código fuente del programa. Es decir, si g invoca a f, f debe declararse antes que g.
- Debe darse soporte a la ejecución de subprogramas recursivos. Un subprograma f es recursivo si dentro de su código se vuelve a invocar a sí mismo. Sin embargo, no se da soporte a la recursividad indirecta. Dos subprogramas f y g guardan una relación de recursividad indirecta si dentro del cuerpo de f se invoca a g, y recíprocamente, dentro de g se invoca a f.

#### Ejemplo recursividad en modulUned

```
MODULE testRecursividad;
VAR a:INTEGER;
      PROCEDURE comparar (VAR x:INTEGER);
      BEGIN
             WRITESTRING("x:");
             WRITEINT(x);
             WRITELN;
             IF x<5 THEN
                   x := x+1;
                    WRITESTRING("x+1:");
                    WRITEINT(x);
                   WRITELN;
                   comparar(x);
             ELSE
                   x := 5;
             END;
             WRITESTRING("Salir subprograma");
             WRITELN;
      END comparar;
BEGIN
      a:=1;
      comparar(a);
      WRITESTRING("a:");
      WRITEINT(a);
      WRITELN;
END testRecursividad;
```

#### Código intermedio:

```
Quadruple - [STARTGLOBAL null, null, null],
Quadruple - [VARGLOBAL A, 0, null],
Quadruple - [PUNTEROGLOBAL T_6, 12, null],
Quadruple - [MV T 0, 1, null],
Quadruple - [MVA T_1, A, null],
Quadruple - [STP T 1, T 0, null],
Quadruple - [STARTSUBPROGRAMA null, null, null],
Quadruple - [MVA T_2, A, null],
Quadruple - [PARAM T_2, null, null],
Quadruple - [CALL L_COMPARAR, null, null],
Quadruple - [WRITESTRING T_3, L_9, null],
Quadruple - [MVP T_4, A, null],
Quadruple - [WRITEINT T 4, null, null],
Quadruple - [WRITELN null, null, null],
Quadruple - [HALT null, null, null],
Quadruple - [ETIQUETA L_COMPARAR, null, null],
Quadruple - [PUNTEROSUBPROGRAMA T 15, 22, null],
Quadruple - [WRITESTRING T 0, L 0, null],
Quadruple - [MVP T_1, X, null],
Quadruple - [WRITEINT T 1, null, null],
Quadruple - [WRITELN null, null, null],
Quadruple - [MVP T_2, X, null],
Quadruple - [MV T_3, 5, null],
Quadruple - [LS T_4, T_2, T_3],
Quadruple - [BRF T_4, L_4, null],
Quadruple - [MVP T 5, X, null],
Quadruple - [MV T 6, 1, null],
Quadruple - [ADD T_7, T_5, T_6],
Quadruple - [MVA T 8, X, null],
Quadruple - [STP T 8, T 7, null],
Quadruple - [WRITESTRING T 9, L 1, null],
Quadruple - [MVP T 10, X, null],
Quadruple - [WRITEINT T 10, null, null],
Quadruple - [WRITELN null, null, null],
Quadruple - [STARTSUBPROGRAMA null, null, null],
Quadruple - [MVA T_11, X, null],
Quadruple - [PARAM T_11, null, null],
Quadruple - [CALL L_COMPARAR, null, null],
Quadruple - [BR L_3, null, null],
Quadruple - [INL L 4, null, null],
Quadruple - [MV T 12, 5, null],
Quadruple - [MVA T_13, X, null],
Quadruple - [STP T 13, T 12, null],
Quadruple - [INL L 3, null, null],
Quadruple - [WRITESTRING T 14, L 5, null],
Quadruple - [WRITELN null, null, null],
Quadruple - [FINSUBPROGRAMA L_FIN_COMPARAR, 6, null],
Quadruple - [CADENA "X:", L 0, null],
Quadruple - [CADENA "X+1:", L 1, null],
Quadruple - [CADENA "SALIR SUBPROGRAMA", L 5, null],
Quadruple - [CADENA "A:", L_9, null]
```

Dentro del Registro de Activación vamos asignando posiciones de memoria a los valores fijos del registro (Valor de retorno, enlace de control,...) variables y temporales. Para acceder a las posiciones de memoria que ocupa el registro de Activación lo haremos a través de direccionamiento relativo al registro índice IX

## R.A. testRecursividad

0	Valor de Retorno	
1	Enlace de Control	
2	Estado maquina	
3	Enlace de Acceso	
4	а	
5	T_0	
6	T_1	
7	T_2	
8	T_3	
9	T_4	
10	T_5	
11	T_6	

R.A. comparar

Valor de Retorno
Enlace de Control
Estado maquina
Enlace de Acceso
х
Dirección de Retorno
T_0
T_1
T_2
T_3
T_4
T_5
T_6
T_7
T_8
T_9
T_10
T_11
T_12
T_13
T_14
T_15

;Quadruple - [STARTGLOBAL null, null, null] ;Quadruple - [WRITESTRING T 3, L 9, null] MOVE .SP, .IX WRSTR/L 9 PUSH #-1 ;Quadruple - [MVP T 4, A, null] PUSH .IX MOVE #-4[.IX], #-9[.IX] PUSH .SR ;Quadruple - [WRITEINT T\_4, null, null] PUSH .IX WRINT #-9[.IX] ;Quadruple - [VARGLOBAL A, 0, null] ;Quadruple - [WRITELN null, null, null] PUSH #0 WRCHAR #10 ;Quadruple - [PUNTEROGLOBAL T\_ 6, 12, null] ;Quadruple - [HALT null, null, null] SUB .IX, #12 HALT MOVE .A, .SP ;Quadruple - [ETIQUETA L COMPARAR, null, null] ;Quadruple - [MV T\_0, 1, null] L COMPARAR: NOP MOVE #1, #-5[.IX] ;Quadruple - [PUNTEROSUBPROGRAMA T\_15, 22, null] ;Quadruple - [MVA T\_1, A, null] SUB .IX, #22 SUB .IX, #4 MOVE .A, .SP MOVE .A, #-6[.IX] ;Quadruple - [WRITESTRING T\_0, L\_0, null] ;Quadruple - [STP T\_1, T\_0, null] WRSTR /L\_0 MOVE #-6[.IX], .R1 ;Quadruple - [MVP T\_1, X, null] MOVE #-5[.IX], [.R1] MOVE #-4[.IX], .R1 ;Quadruple - [STARTSUBPROGRAMA null, null, null] MOVE [.R1], #-7[.IX] MOVE .SP, .RO ;Quadruple - [WRITEINT T 1, null, null] PUSH #-1 WRINT #-7[.IX] PUSH .RO ;Quadruple - [WRITELN null, null, null] PUSH .SR WRCHAR #10 PUSH .IX ;Quadruple - [MVP T\_2, X, null] ;Quadruple - [MVA T\_2, A, null] MOVE #-4[.IX], .R1 SUB .IX, #4 MOVE [.R1], #-8[.IX] MOVE .A, #-7[.IX] ;Quadruple - [MV T\_3, 5, null] ;Quadruple - [PARAM T\_2, null, null] MOVE #5, #-9[.IX] PUSH #-7[.IX] ;Quadruple - [LS T\_4, T\_2, T\_3] ;Quadruple - [CALL L\_COMPARAR, null, null] CMP #-8[.IX], #-9[.IX] MOVE .RO, .IX BN \$5 CALL/L COMPARAR MOVE #0, #-10[.IX] MOVE .IX, .SP BR \$3 MOVE #-3[.IX], .RO MOVE #1, #-10[.IX] MOVE .RO, .IX

;Quadruple - [BRF T\_4, L\_4, null] CMP #0, #-10[.IX] BZ/L 4 ;Quadruple - [MVP T\_5, X, null] MOVE #-4[.IX], .R1 MOVE [.R1], #-11[.IX] ;Quadruple - [MV T\_6, 1, null] MOVE #1, #-12[.IX] ;Quadruple - [ADD T\_7, T\_5, T\_6] ADD #-11[.IX], #-12[.IX] MOVE .A , #-13[.IX] ;Quadruple - [MVA T\_8, X, null] MOVE #-4[.IX], #-14[.IX] ;Quadruple - [STP T\_8, T\_7, null] MOVE #-14[.IX], .R1 MOVE #-13[.IX], [.R1] ;Quadruple - [WRITESTRING T 9, L 1, null] WRSTR /L 1 ;Quadruple - [MVP T\_10, X, null] MOVE #-4[.IX], .R1 MOVE [.R1], #-16[.IX] ;Quadruple - [WRITEINT T\_10, null, null] WRINT #-16[.IX] ;Quadruple - [WRITELN null, null, null] WRCHAR #10 ;Quadruple - [STARTSUBPROGRAMA null, null, null] MOVE .SP, .RO PUSH #-1 PUSH .RO PUSH .SR PUSH .IX ;Quadruple - [MVA T\_11, X, null] MOVE #-4[.IX], #-17[.IX] ;Quadruple - [PARAM T\_11, null, null] PUSH #-17[.IX]

;Quadruple - [CALL L\_COMPARAR, null, null] MOVE .RO, .IX CALL/L COMPARAR MOVE .IX, .SP MOVE #-3[.IX], .RO MOVE .RO, .IX ;Quadruple - [BR L 3, null, null] BR/L 3 ;Quadruple - [INL L 4, null, null] L 4: NOP ;Quadruple - [MV T\_12, 5, null] MOVE #5, #-18[.IX] ;Quadruple - [MVA T\_13, X, null] MOVE #-4[.IX], #-19[.IX] ;Quadruple - [STP T\_13, T\_12, null] MOVE #-19[.IX], .R1 MOVE #-18[.IX], [.R1] ;Quadruple - [INL L\_3, null, null] L 3:NOP ;Quadruple - [WRITESTRING T\_14, L\_5, null] WRSTR/L 5 ;Quadruple - [WRITELN null, null, null] WRCHAR #10 ;Quadruple - [FINSUBPROGRAMA L\_FIN\_COMPARAR, 6, null] L\_FIN\_COMPARAR: NOP SUB .IX, #6 MOVE .A, .SP RFT ;Quadruple - [CADENA "X:", L\_0, null] L 0: DATA "X:" ;Quadruple - [CADENA "X+1:", L\_1, null] L 1: DATA "X+1:" ;Quadruple - [CADENA "SALIR SUBPROGRAMA", L\_5, null] L\_5 : DATA "SALIR SUBPROGRAMA" ;Quadruple - [CADENA "A:", L\_9, null] L\_9: DATA "A:"

## Evolución de la memoria con los Registros de Activación

R.A. comparar 3ª llamada rec. R.A. comparar 2º llamada recursiva

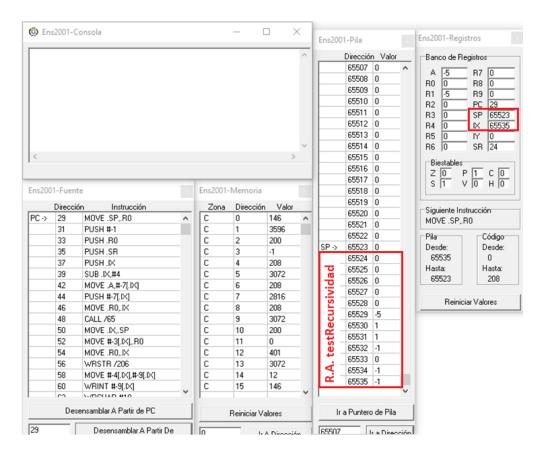
	Valor de Retorno	65535
	Enlace de Control	65534
ä	Estado maquina	65533
<u>.</u>	Enlace de Acceso	65532
- 51	a	65531
<u>ā</u>	T_0	65530
- 2	T_1	65529
R.A. pr. principal	T_2	65528
ď	T_3	65527
œ	T_4	65526
	T_5	65525
	T 6	65524
	Valor de Retorno	65523
	Enlace de Control	65522
	Estado maquina	65521
<u>_</u>	Enlace de Acceso	65520
<u>:0</u>	X	65519
ä	Dirección de retorno	65518
8	T 0	65517
ĕ∣	T_1	65516
R.A. comparar 1ª invocación	T_2	65515
ei.	T_3	65514
- 2	T_4	65513
2	T_5	65512
g	T 6	65511
ᇎ	T_7	65510
ō	T 8	65509
٩١	T 9	65508
₹.	T_10	65507
02	T 11	65506
	T_12 T 13	65505
		65504 65503
	T 15	65502
.A. comparar º llamada recursiva	Valor de Retorno	65501
· <u>v</u>	Enlace de Control	65500
5	Estado maquina	65499
ᇤ	Enlace de Acceso	65498
2 2	x	65497
프	Dirección de retorno	65496
Εğ	T_0	65495
용토	T_1	65494
<u> </u>	T_2	65493
R.A. comparar 1ª llamada rec	T_3	65492
E 4	T_4	65491

T_5	65490
T_6	65489
T_7	65488
T_8	65487
T 9	65486
T_10	65485
T_11	65484
T 12	65483
T_13	65482
T 14	65481
T 15	65480
Valor de Ketorno	654/9
Enlace de Control	65478
Estado maquina	65477
Enlace de Acceso	65476
X	65475
Dirección de retorno	65474
	65474
T_0 T 1	
	65472
T_2	65471
T_3	65470
T_4	65469
T_5	65468
T_6	65467
T_7	65466
T_8	65465
T_9	65464
T_10	65463
T_11	65462
T_12	65461
T_13	65460
T_14	65459
T_15	65458
Valor de Retorno	65457
Enlace de Control	65456
Estado maquina	65455
Enlace de Acceso	65454
X	65453
Dirección de retorno	65452
T_0	65451
T_1	65450
T_2	65449
T_3	65448
T_4	65447
T_5	65446

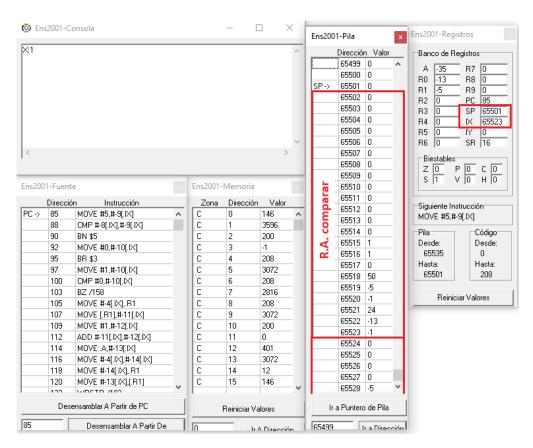
R.A. comparar 4ª llamada recursiva

T_6	65445
T_7	65444
T_8	65443
T_9	65442
T_10	65441
T_11	65440
T_12	65439
T_13	65438
T_14	65437
T 15	65436
Valor de Retorno	65435
Enlace de Control	65434
Estado maquina	65433
Enlace de Acceso	65432
x	65431
Dirección de retorno	65430
T_0	65429
T_1	65428
T_2	65427
T_3	65426
T_4	65425
T_5	65424
T_6	65423
T_7	65422
T_8	65421
T_9	65420
T_10	65419
T_11	65418
T_12	65417
T_13	65416
T_14	65415
T 15	65414
	65413
	66412
	65411
	65410
	65409
	65408
	65407
	65406
	65405
	65404
	65403
	65402
	65401

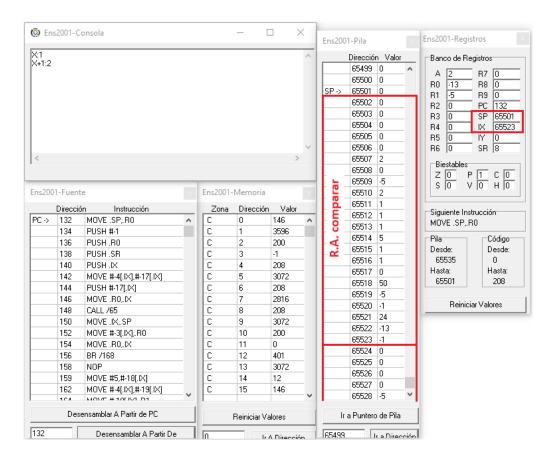
#### R.A. Programa Principal



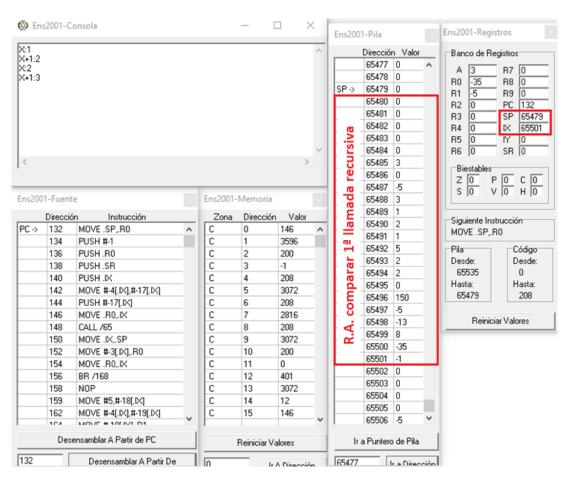
#### R.A. subprograma comparar 1ª invocación



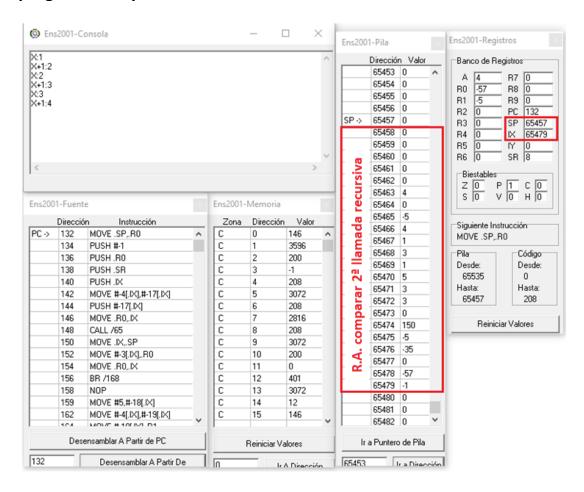
#### R.A. subprograma comparar 1ª invocación



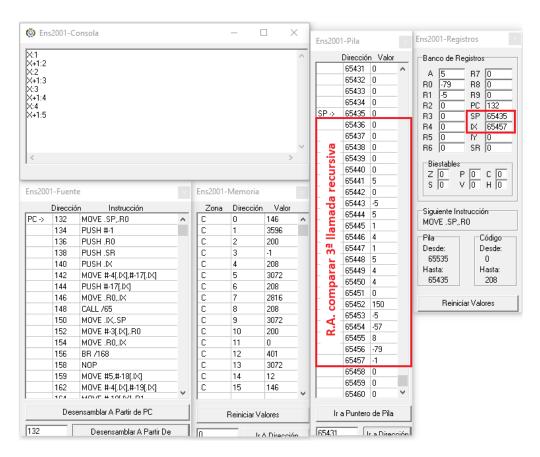
## R.A. subprograma comparar 1ª llamada recursiva



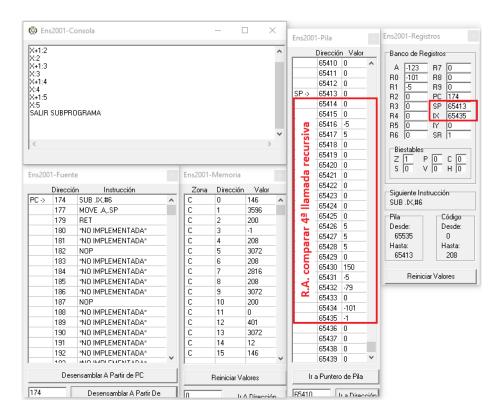
#### R.A. subprograma comparar 2ª llamada recursiva



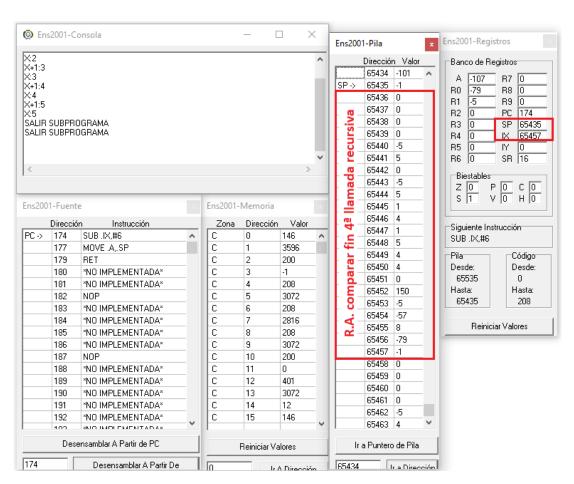
#### R.A. subprograma comparar 3ª llamada recursiva



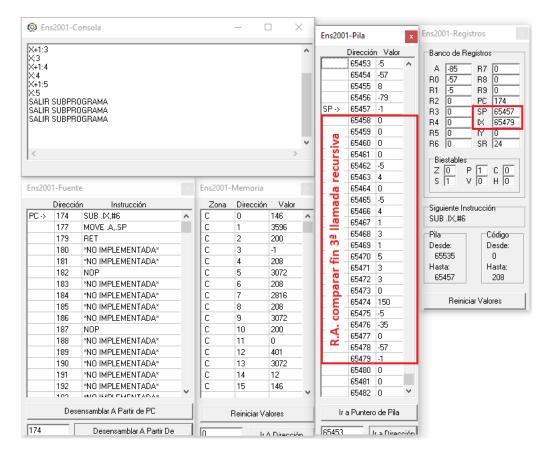
## R.A. subprograma comparar 4ª llamada recursiva



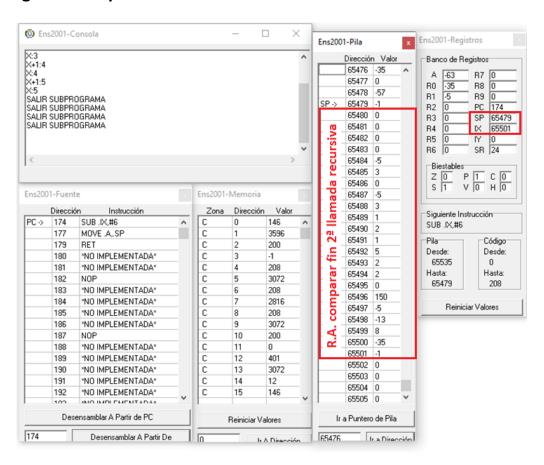
## R.A. subprograma comparar fin 4ª llamada recursiva



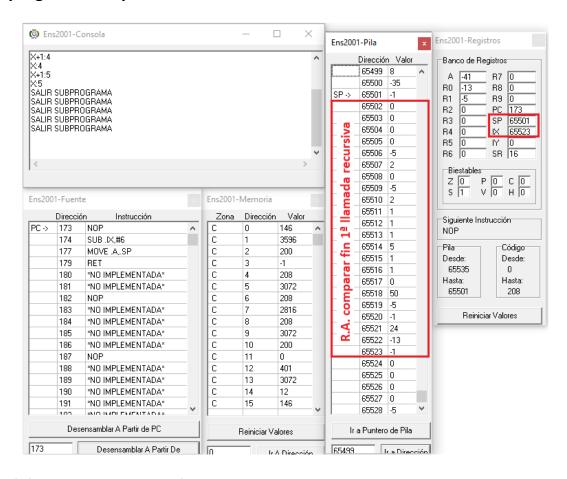
#### R.A. subprograma comparar fin 3ª llamada recursiva



### R.A. subprograma comparar fin 2ª llamada recursiva



#### R.A. subprograma comparar fin 1ª llamada recursiva



#### R.A. Fin del programa principal

