75.14 / 95.48 Lenguajes Formales
Trabajo Práctico 2021

Funcionamiento del intérprete de PL/0
al procesar los archivos BIEN-nn.PL0

Pantalla de inicio:

```
Interprete de PL/0 en Clojure
Trabajo Practico de 75.14/95.48 Lenguajes Formales - 2021

Lista de comandos posibles:
AYUDA: volver a este menu
SALIR: volver al REPL de Clojure
ESCAN <archivo>: mostrar los tokens de un programa escrito en PL/0
VIRTU <archivo>: mostrar la RI de un programa escrito en PL/0
INTER <archivo>: interpretar la RI de un programa escrito en PL/0
```

```
PL/0> escan bien-00.pl0
VAR X , Y ;
PROCEDURE INICIAR ;
CONST Y = 2 ;
PROCEDURE ASIGNAR ;
X := Y ;
CALL ASIGNAR ;
BEGIN
  WRITELN ( '*****************************************************************' ) ;
  WRITELN ( 'Se ingresa un valor, se muestra su doble.' ) ;
  WRITELN ( 'Ejemplo de eclipse de nombres (shadowing) en la asignacion X := Y' ) ;
  WRITELN ( '(la variable global Y es eclipsada por la constante local Y)' ) ;
  WRITELN ( '*****************************************************************' ) ;
  WRITE ( 'NUM=' ) ;
  READLN ( Y ) ;
  CALL INICIAR ;
  WRITELN ( 'NUM*2=' , Y * X ) ;
  WRITELN
END .
```

```
PL/0> virtu bien-00.pl0
0 [JMP 7]
1 [JMP 5]
2 [PFI 2]
3 [POP 0]
4 RET
5 [CAL 2]
6 RET
7 [OUT '*****************************************************************']
8 NL
9 [OUT 'Se ingresa un valor, se muestra su doble.']
10 NL
11 [OUT 'Ejemplo de eclipse de nombres (shadowing) en la asignacion X := Y']
12 NL
13 [OUT '(la variable global Y es eclipsada por la constante local Y)']
14 NL
15 [OUT '*****************************************************************']
16 NL
17 [OUT 'NUM=']
18 [IN 1]
19 [CAL 1]
20 [OUT 'NUM*2=']
21 [PFM 1]
22 [PFM 0]
23 MUL
24 OUT
25 NL
26 NL
27 HLT
```

```
PL/0> inter bien-00.pl0
*****************************************************************
Se ingresa un valor, se muestra su doble.
Ejemplo de eclipse de nombres (shadowing) en la asignacion X := Y
(la variable global Y es eclipsada por la constante local Y)
*****************************************************************
NUM=17
NUM*2=34
```

PL/0> escan bien-01.pl0

```
VAR BASE , EXPO , RESU ;
PROCEDURE POT ;
IF EXPO > 0 THEN
BEGIN
  RESU := RESU * BASE ;
  EXPO := EXPO - 1 ;
  CALL POT
END ;
BEGIN
  WRITELN ( '**************************************************' ) ;
  WRITELN ( 'Se ingresan base y exponente, se muestra la potencia.' ) ;
  WRITELN ( 'Ejemplo de funcion recursiva' ) ;
  WRITELN ( '**************************************************' ) ;
  WRITE ( 'BASE: ' ) ;
  READLN ( BASE ) ;
  WRITE ( 'EXPONENTE: ' ) ;
  READLN ( EXPO ) ;
  RESU := 1 ;
  CALL POT ;
  IF EXPO < 0 THEN RESU := 0 ;
  WRITELN ( 'RESULTADO: ' , RESU ) ;
  WRITELN
END .
```

PL/0> virtu bien-01.pl0

```
0 [JMP 16]
1 [PFM 1]
2 [PFI 0]
3 GT
4 [JC 6]
5 [JMP 15]
6 [PFM 2]
7 [PFM 0]
8 MUL
9 [POP 2]
10 [PFM 1]
11 [PFI 1]
12 SUB
13 [POP 1]
14 [CAL 1]
15 RET
16 [OUT '**************************************************']
17 NL
18 [OUT 'Se ingresan base y exponente, se muestra la potencia.']
19 NL
20 [OUT 'Ejemplo de funcion recursiva']
21 NL
22 [OUT '**************************************************']
23 NL
24 [OUT 'BASE: ']
25 [IN 0]
26 [OUT 'EXPONENTE: ']
27 [IN 1]
28 [PFI 1]
29 [POP 2]
30 [CAL 1]
31 [PFM 1]
32 [PFI 0]
33 LT
34 [JC 36]
35 [JMP 38]
36 [PFI 0]
37 [POP 2]
38 [OUT 'RESULTADO: ']
39 [PFM 2]
40 OUT
41 NL
42 NL
43 HLT
```

PL/0> inter bien-01.pl0

```
**************************************************
Se ingresan base y exponente, se muestra la potencia.
Ejemplo de funcion recursiva
**************************************************
BASE: 16
EXPONENTE: 2
RESULTADO: 256
```

PL/0> escan bien-02.pl0

```
VAR X , Y , Z ;
PROCEDURE MULTIPLICAR ;
VAR A , B ;
BEGIN
  A := X ;
  B := Y ;
  Z := 0 ;
  IF X < 0 THEN A := - A ;
  IF Y < 0 THEN B := - B ;
  WHILE B > 0 DO
  BEGIN
    IF ODD B THEN Z := Z + A ;
    A := A * 2 ;
    B := B / 2
  END ;
  IF X < 0 THEN Z := - Z ;
  IF Y < 0 THEN Z := - Z
END ;
BEGIN
  WRITELN ( '************************************************************' ) ;
  WRITELN ( 'Se ingresan dos valores, se muestra su producto.' ) ;
  WRITELN ( 'Se utiliza el algoritmo de "multiplicacion por duplicacion".' ) ;
  WRITELN ( '(Metodo campesino ruso de multiplicacion)' ) ;
  WRITELN ( '************************************************************' ) ;
  WRITE ( 'X: ' ) ;
  READLN ( X ) ;
  WRITE ( 'Y: ' ) ;
  READLN ( Y ) ;
  CALL MULTIPLICAR ;
  WRITELN ( 'X*Y=' , Z ) ;
  WRITELN ;
END .
```

PL/0> virtu bien-02.pl0

```
0 [JMP 62]
1 [PFM 0]
2 [POP 3]
3 [PFM 1]
4 [POP 4]
5 [PFI 0]
6 [POP 2]
7 [PFM 0]
8 [PFI 0]
9 LT
10 [JC 12]
11 [JMP 15]
12 [PFM 3]
13 NEG
14 [POP 3]
15 [PFM 1]
16 [PFI 0]
17 LT
18 [JC 20]
19 [JMP 23]
20 [PFM 4]
21 NEG
22 [POP 4]
23 [PFM 4]
24 [PFI 0]
25 GT
26 [JC 28]
27 [JMP 45]
28 [PFM 4]
29 ODD
30 [JC 32]
31 [JMP 36]
32 [PFM 2]
33 [PFM 3]
34 ADD
35 [POP 2]
36 [PFM 3]
37 [PFI 2]
38 MUL
39 [POP 3]
40 [PFM 4]
41 [PFI 2]
42 DIV
43 [POP 4]
```

```
44 [JMP 23]
45 [PFM 0]
46 [PFI 0]
47 LT
48 [JC 50]
49 [JMP 53]
50 [PFM 2]
51 NEG
52 [POP 2]
53 [PFM 1]
54 [PFI 0]
55 LT
56 [JC 58]
57 [JMP 61]
58 [PFM 2]
59 NEG
60 [POP 2]
61 RET
62 [OUT '************************************************************']
63 NL
64 [OUT 'Se ingresan dos valores, se muestra su producto.']
65 NL
66 [OUT 'Se utiliza el algoritmo de "multiplicacion por duplicacion".']
67 NL
68 [OUT '(Metodo campesino ruso de multiplicacion)']
69 NL
70 [OUT '************************************************************']
71 NL
72 [OUT 'X: ']
73 [IN 0]
74 [OUT 'Y: ']
75 [IN 1]
76 [CAL 1]
77 [OUT 'X*Y=']
78 [PFM 2]
79 OUT
80 NL
81 NL
82 HLT
```

```
PL/0> inter bien-02.pl0
************************************************************
Se ingresan dos valores, se muestra su producto.
Se utiliza el algoritmo de "multiplicacion por duplicacion".
(Metodo campesino ruso de multiplicacion)
************************************************************
X: 17
Y: 2
X*Y=34
```

```
PL/0> escan bien-03.pl0
```

```
VAR X , Y , Q , R ;
PROCEDURE DIVIDIR ;
VAR V , W ;
BEGIN
  Q := 0 ;
  R := X ;
  IF R < 0 THEN R := - R ;
  W := Y ;
  IF W < 0 THEN W := - W ;
  V := Y ;
  IF V < 0 THEN V := - V ;
  WHILE W <= R DO W := W * 2 ;
  WHILE W > V DO
  BEGIN
    Q := Q * 2 ;
    W := W / 2 ;
    IF W <= R THEN
    BEGIN
      R := R - W ;
      Q := Q + 1
    END
  END ;
  IF X < 0 THEN R := - R ;
  IF X < 0 THEN Q := - Q ;
  IF Y < 0 THEN Q := - Q ;
END ;
PROCEDURE OTRO ;
PROCEDURE DIVIDIR ;
BEGIN
  Q := X / Y ;
  R := X - Y * Q
END ;
CALL DIVIDIR ;
PROCEDURE SALIDA ;
BEGIN
  WRITE ( '  Cociente: ' , Q ) ;
  WRITELN ;
  WRITE ( '  Resto: ' ) ;
  WRITELN ( R ) ;
END ;
BEGIN
  WRITELN ( '************************************************************' ) ;
  WRITELN ( 'Se ingresan dos valores, se muestra su cociente.' ) ;
  WRITELN ( 'Se utilizan dos metodos:' ) ;
  WRITELN ( 'Metodo 1: el algoritmo "desplazar y restar" (shift-subtract)' ) ;
  WRITELN ( 'Metodo 2: el operador / (division entera) provisto por PL/0.' ) ;
  WRITELN ( 'Cada CALL DIVIDIR se refiere a un procedimiento distinto.' ) ;
  WRITELN ( '************************************************************' ) ;
  WRITE ( 'Dividendo: ' ) ;
  READLN ( X ) ;
  WRITE ( 'Divisor: ' ) ;
  READLN ( Y ) ;
  WRITELN ;
  IF Y <> 0 THEN
  BEGIN
    WRITE ( 'Metodo 1' ) ;
    WRITELN ;
    CALL DIVIDIR ;
    CALL SALIDA ;
    WRITELN ;
    WRITE ( 'Metodo 2' ) ;
    WRITELN ;
    CALL OTRO ;
    CALL SALIDA ;
  END ;
  WRITELN
END .
```

```
PL/0> virtu bien-03.pl0
```

```
0 [JMP 118]
1 [PFI 0]
2 [POP 2]
3 [PFM 0]
4 [POP 3]
5 [PFM 3]
6 [PFI 0]
7 LT
8 [JC 10]
```

```
9 [JMP 13]
10 [PFM 3]
11 NEG
12 [POP 3]
13 [PFM 1]
14 [POP 5]
15 [PFM 5]
16 [PFI 0]
17 LT
18 [JC 20]
19 [JMP 23]
20 [PFM 5]
21 NEG
22 [POP 5]
23 [PFM 1]
24 [POP 4]
25 [PFM 4]
26 [PFI 0]
27 LT
28 [JC 30]
29 [JMP 33]
30 [PFM 4]
31 NEG
32 [POP 4]
33 [PFM 5]
34 [PFM 3]
35 LTE
36 [JC 38]
37 [JMP 43]
38 [PFM 5]
39 [PFI 2]
40 MUL
41 [POP 5]
42 [JMP 33]
43 [PFM 5]
44 [PFM 4]
45 GT
46 [JC 48]
47 [JMP 70]
48 [PFM 2]
49 [PFI 2]
50 MUL
51 [POP 2]
52 [PFM 5]
53 [PFI 2]
54 DIV
55 [POP 5]
56 [PFM 5]
57 [PFM 3]
58 LTE
59 [JC 61]
60 [JMP 69]
61 [PFM 3]
62 [PFM 5]
63 SUB
64 [POP 3]
65 [PFM 2]
66 [PFI 1]
67 ADD
68 [POP 2]
69 [JMP 43]
70 [PFM 0]
71 [PFI 0]
72 LT
73 [JC 75]
74 [JMP 78]
75 [PFM 3]
76 NEG
77 [POP 3]
78 [PFM 0]
79 [PFI 0]
80 LT
81 [JC 83]
82 [JMP 86]
83 [PFM 2]
84 NEG
85 [POP 2]
86 [PFM 1]
```

```
87 [PFI 0]
88 LT
89 [JC 91]
90 [JMP 94]
91 [PFM 2]
92 NEG
93 [POP 2]
94 RET
95 [JMP 107]
96 [PFM 0]
97 [PFM 1]
98 DIV
99 [POP 2]
100 [PFM 0]
101 [PFM 1]
102 [PFM 2]
103 MUL
104 SUB
105 [POP 3]
106 RET
107 [CAL 96]
108 RET
109 [OUT '  Cociente: ']
110 [PFM 2]
111 OUT
112 NL
113 [OUT '  Resto: ']
114 [PFM 3]
115 OUT
116 NL
117 RET
118 [OUT '************************************************************']
119 NL
120 [OUT 'Se ingresan dos valores, se muestra su cociente.']
121 NL
122 [OUT 'Se utilizan dos metodos:']
123 NL
124 [OUT 'Metodo 1: el algoritmo "desplazar y restar" (shift-subtract)']
125 NL
126 [OUT 'Metodo 2: el operador / (division entera) provisto por PL/0.']
127 NL
128 [OUT 'Cada CALL DIVIDIR se refiere a un procedimiento distinto.']
129 NL
130 [OUT '************************************************************']
131 NL
132 [OUT 'Dividendo: ']
133 [IN 0]
134 [OUT 'Divisor: ']
135 [IN 1]
136 NL
137 [PFM 1]
138 [PFI 0]
139 NEQ
140 [JC 142]
141 [JMP 151]
142 [OUT 'Metodo 1']
143 NL
144 [CAL 1]
145 [CAL 109]
146 NL
147 [OUT 'Metodo 2']
148 NL
149 [CAL 95]
150 [CAL 109]
151 NL
152 HLT
```

```
PL/0> inter bien-03.pl0
************************************************************
Se ingresan dos valores, se muestra su cociente.
Se utilizan dos metodos:
Metodo 1: el algoritmo "desplazar y restar" (shift-subtract)
Metodo 2: el operador / (division entera) provisto por PL/0.
Cada CALL DIVIDIR se refiere a un procedimiento distinto.
************************************************************
Dividendo: 34
Divisor: 4

Metodo 1
```

```
  Cociente: 8
  Resto: 2

Metodo 2
  Cociente: 8
  Resto: 2
```

```
PL/0> escan bien-04.pl0
```

```
VAR X , Y , Z ;
PROCEDURE MCD ;
VAR F , G ;
BEGIN
  F := X ;
  G := Y ;
  WHILE F <> G DO
  BEGIN
    IF F < G THEN G := G - F ;
    IF G < F THEN F := F - G
  END ;
  Z := F
END ;
BEGIN
  WRITELN ( '************************************************************' ) ;
  WRITELN ( 'Se ingresan dos valores, se muestra su maximo comun divisor.' ) ;
  WRITELN ( 'Se utiliza el algoritmo de Euclides.' ) ;
  WRITELN ( '************************************************************' ) ;
  WRITE ( 'X: ' ) ;
  READLN ( X ) ;
  IF X > 0 THEN
  BEGIN
    WRITE ( 'Y: ' ) ;
    READLN ( Y ) ;
    IF Y > 0 THEN
    BEGIN
      CALL MCD ;
      WRITELN ( 'MCD: ' , Z )
    END
  END ;
  WRITELN
END .
```

```
PL/0> virtu bien-04.pl0
```

```
0 [JMP 32]
1 [PFM 0]
2 [POP 3]
3 [PFM 1]
4 [POP 4]
5 [PFM 3]
6 [PFM 4]
7 NEQ
8 [JC 10]
9 [JMP 29]
10 [PFM 3]
11 [PFM 4]
12 LT
13 [JC 15]
14 [JMP 19]
15 [PFM 4]
16 [PFM 3]
17 SUB
18 [POP 4]
19 [PFM 4]
20 [PFM 3]
21 LT
22 [JC 24]
23 [JMP 28]
24 [PFM 3]
25 [PFM 4]
26 SUB
27 [POP 3]
28 [JMP 5]
29 [PFM 3]
30 [POP 2]
31 RET
32 [OUT '************************************************************']
33 NL
34 [OUT 'Se ingresan dos valores, se muestra su maximo comun divisor.']
35 NL
36 [OUT 'Se utiliza el algoritmo de Euclides.']
37 NL
38 [OUT '************************************************************']
39 NL
40 [OUT 'X: ']
41 [IN 0]
42 [PFM 0]
43 [PFI 0]
```

```
44 GT
45 [JC 47]
46 [JMP 59]
47 [OUT 'Y: ']
48 [IN 1]
49 [PFM 1]
50 [PFI 0]
51 GT
52 [JC 54]
53 [JMP 59]
54 [CAL 1]
55 [OUT 'MCD: ']
56 [PFM 2]
57 OUT
58 NL
59 NL
60 HLT
```

```
PL/0> inter bien-04.pl0
************************************************************
Se ingresan dos valores, se muestra su maximo comun divisor.
Se utiliza el algoritmo de Euclides.
************************************************************
X: 16
Y: 24
MCD: 8
```

```
PL/0> escan bien-05.pl0
```

```
VAR R , N ;
PROCEDURE INICIALIZAR ;
CONST UNO = 1 ;
R := - ( - UNO ) ;
PROCEDURE RAIZ ;
BEGIN
  CALL INICIALIZAR ;
  WHILE R * R < N DO R := R + 1
END ;
BEGIN
  WRITELN ( '************************************************' ) ;
  WRITELN ( 'Se ingresa un valor, se muestra su raiz cuadrada o' ) ;
  WRITELN ( 'el intervalo al que pertenece, si esta no es entera.' ) ;
  WRITELN ( 'Se utiliza el algoritmo de busqueda lineal.' ) ;
  WRITELN ( '************************************************' ) ;
  WRITE ( 'N: ' ) ;
  READLN ( N ) ;
  WRITE ( 'RAIZ CUADRADA DE ' , N , ': ' ) ;
  IF N < 0 THEN WRITE ( 'ERROR' ) ;
  IF N = 0 THEN WRITE ( 0 ) ;
  IF N > 0 THEN
  BEGIN
    CALL RAIZ ;
    IF R * R <> N THEN WRITE ( R - 1 , '..' ) ;
    WRITE ( R ) ;
  END ;
  WRITELN ;
  WRITELN ;
END .
```

```
PL/0> virtu bien-05.pl0
```

```
0 [JMP 20]
1 [PFI 1]
2 NEG
3 NEG
4 [POP 0]
5 RET
6 [CAL 1]
7 [PFM 0]
8 [PFM 0]
9 MUL
10 [PFM 1]
11 LT
12 [JC 14]
13 [JMP 19]
14 [PFM 0]
15 [PFI 1]
16 ADD
17 [POP 0]
18 [JMP 7]
19 RET
20 [OUT '************************************************']
21 NL
22 [OUT 'Se ingresa un valor, se muestra su raiz cuadrada o']
23 NL
24 [OUT 'el intervalo al que pertenece, si esta no es entera.']
25 NL
26 [OUT 'Se utiliza el algoritmo de busqueda lineal.']
27 NL
28 [OUT '************************************************']
29 NL
30 [OUT 'N: ']
31 [IN 1]
32 [OUT 'RAIZ CUADRADA DE ']
33 [PFM 1]
34 OUT
35 [OUT ': ']
36 [PFM 1]
37 [PFI 0]
38 LT
39 [JC 41]
40 [JMP 42]
41 [OUT 'ERROR']
42 [PFM 1]
43 [PFI 0]
44 EQ
45 [JC 47]
46 [JMP 49]
```

```
47 [PFI 0]
48 OUT
49 [PFM 1]
50 [PFI 0]
51 GT
52 [JC 54]
53 [JMP 69]
54 [CAL 6]
55 [PFM 0]
56 [PFM 0]
57 MUL
58 [PFM 1]
59 NEQ
60 [JC 62]
61 [JMP 67]
62 [PFM 0]
63 [PFI 1]
64 SUB
65 OUT
66 [OUT '..']
67 [PFM 0]
68 OUT
69 NL
70 NL
71 HLT
```

```
PL/0> inter bien-05.pl0
**************************************************
Se ingresa un valor, se muestra su raiz cuadrada o
el intervalo al que pertenece, si esta no es entera.
Se utiliza el algoritmo de busqueda lineal.
**************************************************
N: 81
RAIZ CUADRADA DE 81: 9
```

```
PL/0> inter bien-05.pl0
**************************************************
Se ingresa un valor, se muestra su raiz cuadrada o
el intervalo al que pertenece, si esta no es entera.
Se utiliza el algoritmo de busqueda lineal.
**************************************************
N: 83
RAIZ CUADRADA DE 83: 9..10
```

```
PL/0> escan bien-06.pl0
VAR IMPORTE , BILLETE , VUELTO , USDOLARES , CENTAVOS , VUELTOUSDOLARES , VUELTOCENTAVOS ,
IMPORTEOK , CENTAVOSOK , USDOLARESOK , BILLETEOK , VUELTOOK ;
PROCEDURE ERROR ;
BEGIN
  WRITELN ( 'VALOR FUERA DE RANGO!' )
END ;
BEGIN
  WRITELN ( '************************************' ) ;
  WRITELN ( '   VUELTO PARA IMPORTES EN DOLARES   ' ) ;
  WRITELN ( 'PAGADOS CON UN BILLETE EN EL FREESHOP' ) ;
  WRITELN ( '************************************' ) ;
  IMPORTEOK := - 1 ;
  WHILE IMPORTEOK <> 0 DO
  BEGIN
    WRITELN ( 'IMPORTE (min US$ 0.01 y max US$ 100.00)' ) ;
    CENTAVOSOK := - 1 ;
    WHILE CENTAVOSOK <> 0 DO
    BEGIN
      WRITE ( 'CENTAVOS DE US$: ' ) ;
      READLN ( CENTAVOS ) ;
      CENTAVOSOK := 0 ;
      IF CENTAVOS < 0 THEN CENTAVOSOK := - 1 ;
      IF CENTAVOS > 99 THEN CENTAVOSOK := - 1 ;
      IF CENTAVOSOK <> 0 THEN CALL ERROR
    END ;
    USDOLARESOK := - 1 ;
    WHILE USDOLARESOK <> 0 DO
    BEGIN
      WRITE ( 'DOLARES: ' ) ;
      READLN ( USDOLARES ) ;
      USDOLARESOK := 0 ;
      IF USDOLARES < 0 THEN USDOLARESOK := - 1 ;
      IF USDOLARES > 100 THEN USDOLARESOK := - 1 ;
      IF USDOLARESOK <> 0 THEN CALL ERROR
    END ;
    WRITE ( 'IMPORTE: US$ ' , USDOLARES , '.' ) ;
    IF CENTAVOS < 10 THEN WRITE ( '0' ) ;
    WRITELN ( CENTAVOS ) ;
    IMPORTE := USDOLARES * 100 + CENTAVOS ;
    IMPORTEOK := 0 ;
    IF IMPORTE < 1 THEN IMPORTEOK := - 1 ;
    IF IMPORTE > 10000 THEN IMPORTEOK := - 1 ;
    IF IMPORTEOK <> 0 THEN CALL ERROR
  END ;
  VUELTOOK := - 1 ;
  WHILE VUELTOOK <> 0 DO
  BEGIN
    BILLETEOK := - 1 ;
    WHILE BILLETEOK <> 0 DO
    BEGIN
      WRITE ( 'BILLETE (min US$ 1 y max US$ 100): US$ ' ) ;
      READLN ( BILLETE ) ;
      BILLETEOK := - 1 ;
      IF BILLETE = 1 THEN BILLETEOK := 0 ;
      IF BILLETE = 2 THEN BILLETEOK := 0 ;
      IF BILLETE = 5 THEN BILLETEOK := 0 ;
      IF BILLETE = 10 THEN BILLETEOK := 0 ;
      IF BILLETE = 20 THEN BILLETEOK := 0 ;
      IF BILLETE = 50 THEN BILLETEOK := 0 ;
      IF BILLETE = 100 THEN BILLETEOK := 0 ;
      IF BILLETEOK <> 0 THEN WRITELN ( 'BILLETE INEXISTENTE!' )
    END ;
    VUELTO := BILLETE * 100 - IMPORTE ;
    VUELTOOK := 0 ;
    IF VUELTO < 0 THEN
    BEGIN
      VUELTOOK := - 1 ;
      WRITELN ( 'BILLETE INSUFICIENTE PARA EL PAGO!' )
    END
  END ;
  VUELTOUSDOLARES := VUELTO / 100 ;
  VUELTOCENTAVOS := VUELTO - VUELTOUSDOLARES * 100 ;
  WRITE ( 'Su vuelto: US$ ' , VUELTOUSDOLARES , '.' ) ;
  IF VUELTOCENTAVOS < 10 THEN WRITE ( '0' ) ;
  WRITELN ( VUELTOCENTAVOS ) ;
  WRITELN ;
  IF VUELTO >= 5000 THEN
```

```
  BEGIN
    WRITELN ( '1 billete de US$ 50' ) ;
    VUELTO := VUELTO - 5000
  END ;
  IF VUELTO >= 2000 THEN
  BEGIN
    IF VUELTO / 2000 = 1 THEN WRITELN ( '1 billete de US$ 20' ) ;
    IF VUELTO / 2000 <> 1 THEN WRITELN ( '2 billetes de US$ 20' ) ;
    VUELTO := VUELTO - VUELTO / 2000 * 2000
  END ;
  IF VUELTO >= 1000 THEN
  BEGIN
    WRITELN ( '1 billete de US$ 10' ) ;
    VUELTO := VUELTO - 1000
  END ;
  IF VUELTO >= 500 THEN
  BEGIN
    WRITELN ( '1 billete de US$ 5' ) ;
    VUELTO := VUELTO - 500
  END ;
  IF VUELTO >= 200 THEN
  BEGIN
    IF VUELTO / 200 = 1 THEN WRITELN ( '1 billete de US$ 2' ) ;
    IF VUELTO / 200 <> 1 THEN WRITELN ( '2 billetes de US$ 2' ) ;
    VUELTO := VUELTO - VUELTO / 200 * 200
  END ;
  IF VUELTO >= 100 THEN
  BEGIN
    WRITELN ( '1 billete de US$ 1' ) ;
    VUELTO := VUELTO - 100
  END ;
  IF VUELTO >= 50 THEN
  BEGIN
    WRITELN ( '1 moneda de 50 centavos de US$' ) ;
    VUELTO := VUELTO - 50
  END ;
  IF VUELTO >= 25 THEN
  BEGIN
    WRITELN ( '1 moneda de 25 centavos de US$' ) ;
    VUELTO := VUELTO - 25
  END ;
  IF VUELTO >= 10 THEN
  BEGIN
    IF VUELTO / 10 = 1 THEN WRITELN ( '1 moneda de 10 centavos de US$' ) ;
    IF VUELTO / 10 <> 1 THEN WRITELN ( '2 monedas de 10 centavos de US$' ) ;
    VUELTO := VUELTO - VUELTO / 10 * 10
  END ;
  IF VUELTO >= 5 THEN
  BEGIN
    WRITELN ( '1 moneda de 5 centavos de US$' ) ;
    VUELTO := VUELTO - 5
  END ;
  IF VUELTO > 1 THEN WRITELN ( VUELTO , ' monedas de 1 centavo de US$' ) ;
  IF VUELTO = 1 THEN WRITELN ( '1 moneda de 1 centavo de US$' ) ;
  WRITELN
END .
```

```
PL/0> virtu bien-06.pl0
```

```
0 [JMP 4]
1 [OUT 'VALOR FUERA DE RANGO!']
2 NL
3 RET
4 [OUT '************************************']
5 NL
6 [OUT '   VUELTO PARA IMPORTES EN DOLARES   ']
7 NL
8 [OUT 'PAGADOS CON UN BILLETE EN EL FREESHOP']
9 NL
10 [OUT '************************************']
11 NL
12 [PFI 1]
13 NEG
14 [POP 7]
15 [PFM 7]
16 [PFI 0]
17 NEQ
18 [JC 20]
19 [JMP 136]
20 [OUT 'IMPORTE (min US$ 0.01 y max US$ 100.00)']
```

```
21 NL
22 [PFI 1]
23 NEG
24 [POP 8]
25 [PFM 8]
26 [PFI 0]
27 NEQ
28 [JC 30]
29 [JMP 57]
30 [OUT 'CENTAVOS DE US$: ']
31 [IN 4]
32 [PFI 0]
33 [POP 8]
34 [PFM 4]
35 [PFI 0]
36 LT
37 [JC 39]
38 [JMP 42]
39 [PFI 1]
40 NEG
41 [POP 8]
42 [PFM 4]
43 [PFI 99]
44 GT
45 [JC 47]
46 [JMP 50]
47 [PFI 1]
48 NEG
49 [POP 8]
50 [PFM 8]
51 [PFI 0]
52 NEQ
53 [JC 55]
54 [JMP 56]
55 [CAL 1]
56 [JMP 25]
57 [PFI 1]
58 NEG
59 [POP 9]
60 [PFM 9]
61 [PFI 0]
62 NEQ
63 [JC 65]
64 [JMP 92]
65 [OUT 'DOLARES: ']
66 [IN 3]
67 [PFI 0]
68 [POP 9]
69 [PFM 3]
70 [PFI 0]
71 LT
72 [JC 74]
73 [JMP 77]
74 [PFI 1]
75 NEG
76 [POP 9]
77 [PFM 3]
78 [PFI 100]
79 GT
80 [JC 82]
81 [JMP 85]
82 [PFI 1]
83 NEG
84 [POP 9]
85 [PFM 9]
86 [PFI 0]
87 NEQ
88 [JC 90]
89 [JMP 91]
90 [CAL 1]
91 [JMP 60]
92 [OUT 'IMPORTE: US$ ']
93 [PFM 3]
94 OUT
95 [OUT '.']
96 [PFM 4]
97 [PFI 10]
98 LT
```

```
99 [JC 101]
100 [JMP 102]
101 [OUT '0']
102 [PFM 4]
103 OUT
104 NL
105 [PFM 3]
106 [PFI 100]
107 MUL
108 [PFM 4]
109 ADD
110 [POP 0]
111 [PFI 0]
112 [POP 7]
113 [PFM 0]
114 [PFI 1]
115 LT
116 [JC 118]
117 [JMP 121]
118 [PFI 1]
119 NEG
120 [POP 7]
121 [PFM 0]
122 [PFI 10000]
123 GT
124 [JC 126]
125 [JMP 129]
126 [PFI 1]
127 NEG
128 [POP 7]
129 [PFM 7]
130 [PFI 0]
131 NEQ
132 [JC 134]
133 [JMP 135]
134 [CAL 1]
135 [JMP 15]
136 [PFI 1]
137 NEG
138 [POP 11]
139 [PFM 11]
140 [PFI 0]
141 NEQ
142 [JC 144]
143 [JMP 233]
144 [PFI 1]
145 NEG
146 [POP 10]
147 [PFM 10]
148 [PFI 0]
149 NEQ
150 [JC 152]
151 [JMP 214]
152 [OUT 'BILLETE (min US$ 1 y max US$ 100): US$ ']
153 [IN 1]
154 [PFI 1]
155 NEG
156 [POP 10]
157 [PFM 1]
158 [PFI 1]
159 EQ
160 [JC 162]
161 [JMP 164]
162 [PFI 0]
163 [POP 10]
164 [PFM 1]
165 [PFI 2]
166 EQ
167 [JC 169]
168 [JMP 171]
169 [PFI 0]
170 [POP 10]
171 [PFM 1]
172 [PFI 5]
173 EQ
174 [JC 176]
175 [JMP 178]
176 [PFI 0]
```

```
177 [POP 10]
178 [PFM 1]
179 [PFI 10]
180 EQ
181 [JC 183]
182 [JMP 185]
183 [PFI 0]
184 [POP 10]
185 [PFM 1]
186 [PFI 20]
187 EQ
188 [JC 190]
189 [JMP 192]
190 [PFI 0]
191 [POP 10]
192 [PFM 1]
193 [PFI 50]
194 EQ
195 [JC 197]
196 [JMP 199]
197 [PFI 0]
198 [POP 10]
199 [PFM 1]
200 [PFI 100]
201 EQ
202 [JC 204]
203 [JMP 206]
204 [PFI 0]
205 [POP 10]
206 [PFM 10]
207 [PFI 0]
208 NEQ
209 [JC 211]
210 [JMP 213]
211 [OUT 'BILLETE INEXISTENTE!']
212 NL
213 [JMP 147]
214 [PFM 1]
215 [PFI 100]
216 MUL
217 [PFM 0]
218 SUB
219 [POP 2]
220 [PFI 0]
221 [POP 11]
222 [PFM 2]
223 [PFI 0]
224 LT
225 [JC 227]
226 [JMP 232]
227 [PFI 1]
228 NEG
229 [POP 11]
230 [OUT 'BILLETE INSUFICIENTE PARA EL PAGO!']
231 NL
232 [JMP 139]
233 [PFM 2]
234 [PFI 100]
235 DIV
236 [POP 5]
237 [PFM 2]
238 [PFM 5]
239 [PFI 100]
240 MUL
241 SUB
242 [POP 6]
243 [OUT 'Su vuelto: US$ ']
244 [PFM 5]
245 OUT
246 [OUT '.']
247 [PFM 6]
248 [PFI 10]
249 LT
250 [JC 252]
251 [JMP 253]
252 [OUT '0']
253 [PFM 6]
254 OUT
```

```
255 NL
256 NL
257 [PFM 2]
258 [PFI 5000]
259 GTE
260 [JC 262]
261 [JMP 268]
262 [OUT '1 billete de US$ 50']
263 NL
264 [PFM 2]
265 [PFI 5000]
266 SUB
267 [POP 2]
268 [PFM 2]
269 [PFI 2000]
270 GTE
271 [JC 273]
272 [JMP 299]
273 [PFM 2]
274 [PFI 2000]
275 DIV
276 [PFI 1]
277 EQ
278 [JC 280]
279 [JMP 282]
280 [OUT '1 billete de US$ 20']
281 NL
282 [PFM 2]
283 [PFI 2000]
284 DIV
285 [PFI 1]
286 NEQ
287 [JC 289]
288 [JMP 291]
289 [OUT '2 billetes de US$ 20']
290 NL
291 [PFM 2]
292 [PFM 2]
293 [PFI 2000]
294 DIV
295 [PFI 2000]
296 MUL
297 SUB
298 [POP 2]
299 [PFM 2]
300 [PFI 1000]
301 GTE
302 [JC 304]
303 [JMP 310]
304 [OUT '1 billete de US$ 10']
305 NL
306 [PFM 2]
307 [PFI 1000]
308 SUB
309 [POP 2]
310 [PFM 2]
311 [PFI 500]
312 GTE
313 [JC 315]
314 [JMP 321]
315 [OUT '1 billete de US$ 5']
316 NL
317 [PFM 2]
318 [PFI 500]
319 SUB
320 [POP 2]
321 [PFM 2]
322 [PFI 200]
323 GTE
324 [JC 326]
325 [JMP 352]
326 [PFM 2]
327 [PFI 200]
328 DIV
329 [PFI 1]
330 EQ
331 [JC 333]
332 [JMP 335]
```

```
333 [OUT '1 billete de US$ 2']
334 NL
335 [PFM 2]
336 [PFI 200]
337 DIV
338 [PFI 1]
339 NEQ
340 [JC 342]
341 [JMP 344]
342 [OUT '2 billetes de US$ 2']
343 NL
344 [PFM 2]
345 [PFM 2]
346 [PFI 200]
347 DIV
348 [PFI 200]
349 MUL
350 SUB
351 [POP 2]
352 [PFM 2]
353 [PFI 100]
354 GTE
355 [JC 357]
356 [JMP 363]
357 [OUT '1 billete de US$ 1']
358 NL
359 [PFM 2]
360 [PFI 100]
361 SUB
362 [POP 2]
363 [PFM 2]
364 [PFI 50]
365 GTE
366 [JC 368]
367 [JMP 374]
368 [OUT '1 moneda de 50 centavos de US$']
369 NL
370 [PFM 2]
371 [PFI 50]
372 SUB
373 [POP 2]
374 [PFM 2]
375 [PFI 25]
376 GTE
377 [JC 379]
378 [JMP 385]
379 [OUT '1 moneda de 25 centavos de US$']
380 NL
381 [PFM 2]
382 [PFI 25]
383 SUB
384 [POP 2]
385 [PFM 2]
386 [PFI 10]
387 GTE
388 [JC 390]
389 [JMP 416]
390 [PFM 2]
391 [PFI 10]
392 DIV
393 [PFI 1]
394 EQ
395 [JC 397]
396 [JMP 399]
397 [OUT '1 moneda de 10 centavos de US$']
398 NL
399 [PFM 2]
400 [PFI 10]
401 DIV
402 [PFI 1]
403 NEQ
404 [JC 406]
405 [JMP 408]
406 [OUT '2 monedas de 10 centavos de US$']
407 NL
408 [PFM 2]
409 [PFM 2]
410 [PFI 10]
```

```
411 DIV
412 [PFI 10]
413 MUL
414 SUB
415 [POP 2]
416 [PFM 2]
417 [PFI 5]
418 GTE
419 [JC 421]
420 [JMP 427]
421 [OUT '1 moneda de 5 centavos de US$']
422 NL
423 [PFM 2]
424 [PFI 5]
425 SUB
426 [POP 2]
427 [PFM 2]
428 [PFI 1]
429 GT
430 [JC 432]
431 [JMP 436]
432 [PFM 2]
433 OUT
434 [OUT ' monedas de 1 centavo de US$']
435 NL
436 [PFM 2]
437 [PFI 1]
438 EQ
439 [JC 441]
440 [JMP 443]
441 [OUT '1 moneda de 1 centavo de US$']
442 NL
443 NL
444 HLT
```

```
PL/0> inter bien-06.pl0
************************************
   VUELTO PARA IMPORTES EN DOLARES
PAGADOS CON UN BILLETE EN EL FREESHOP
************************************
IMPORTE (min US$ 0.01 y max US$ 100.00)
CENTAVOS DE US$: 32
DOLARES: 51
IMPORTE: US$ 51.32
BILLETE (min US$ 1 y max US$ 100): US$ 100
Su vuelto: US$ 48.68

2 billetes de US$ 20
1 billete de US$ 5
1 billete de US$ 2
1 billete de US$ 1
1 moneda de 50 centavos de US$
1 moneda de 10 centavos de US$
1 moneda de 5 centavos de US$
3 monedas de 1 centavo de US$
```

```
PL/0> escan bien-07.pl0
VAR K ;
PROCEDURE P ;
PROCEDURE COMA ;
BEGIN
  WRITE ( ',' ) ;
  K := K + 1 ;
  CALL P
END ;
BEGIN
  IF K < 10 THEN
  BEGIN
    WRITE ( K ) ;
    CALL COMA ;
  END
END ;
BEGIN
  WRITELN ( '************************************************************' ) ;
  WRITELN ( 'Se muestran los numeros del 1 al 10, separados por comas.' ) ;
  WRITELN ( 'El procedimiento P llama al procedimiento coma y viceversa.' ) ;
  WRITELN ( 'El problema de la inexistencia de "forward declarations"' ) ;
  WRITELN ( 'se soluciona mediante el anidamiento de los procedimientos.' ) ;
  WRITELN ( '************************************************************' ) ;
  K := 1 ;
  CALL P ;
  WRITELN ( 10 ) ;
  WRITELN
END .
PL/0> virtu bien-07.pl0
0 [JMP 18]
1 [JMP 9]
2 [OUT ',']
3 [PFM 0]
4 [PFI 1]
5 ADD
6 [POP 0]
7 [CAL 1]
8 RET
9 [PFM 0]
10 [PFI 10]
11 LT
12 [JC 14]
13 [JMP 17]
14 [PFM 0]
15 OUT
16 [CAL 2]
17 RET
18 [OUT '************************************************************']
19 NL
20 [OUT 'Se muestran los numeros del 1 al 10, separados por comas.']
21 NL
22 [OUT 'El procedimiento P llama al procedimiento coma y viceversa.']
23 NL
24 [OUT 'El problema de la inexistencia de "forward declarations"']
25 NL
26 [OUT 'se soluciona mediante el anidamiento de los procedimientos.']
27 NL
28 [OUT '************************************************************']
29 NL
30 [PFI 1]
31 [POP 0]
32 [CAL 1]
33 [PFI 10]
34 OUT
35 NL
36 NL
37 HLT
PL/0> inter bien-07.pl0
************************************************************
Se muestran los numeros del 1 al 10, separados por comas.
El procedimiento P llama al procedimiento coma y viceversa.
El problema de la inexistencia de "forward declarations"
se soluciona mediante el anidamiento de los procedimientos.
************************************************************
1,2,3,4,5,6,7,8,9,10
```

```
PL/0> escan bien-08.pl0
```

```
CONST N = 20 ;
VAR A , B , C ;
PROCEDURE TRIANGULO ;
VAR A , B ;
BEGIN
  WRITELN ;
  A := 15 ;
  WHILE A > 0 DO
  BEGIN
    B := 0 ;
    WHILE B < A DO
    BEGIN
      WRITE ( '*' ) ;
      B := B + 1
    END ;
    WRITELN ;
    A := A - 1 ;
  END
END ;
BEGIN
  WRITELN ( '************************************************' ) ;
  WRITELN ( 'ARTE ASCII EN PL/0.' ) ;
  WRITELN ( 'DOS SECUENCIAS ENTERAS Y UN TRIANGULO RECTANGULO...' ) ;
  WRITELN ( 'ANONIMO (S. XX)' ) ;
  WRITELN ( '************************************************' ) ;
  A := 1 ;
  WHILE A <= N DO
  BEGIN
    WRITE ( A , ' ' ) ;
    A := A + 1
  END ;
  CALL TRIANGULO ;
  B := - N ;
  C := 0 ;
  WHILE B < C DO
  BEGIN
    WRITE ( B , ' ' ) ;
    B := B + 1
  END ;
  WRITELN ;
  WRITELN
END .
```

```
PL/0> virtu bien-08.pl0
```

```
0 [JMP 29]
1 NL
2 [PFI 15]
3 [POP 3]
4 [PFM 3]
5 [PFI 0]
6 GT
7 [JC 9]
8 [JMP 28]
9 [PFI 0]
10 [POP 4]
11 [PFM 4]
12 [PFM 3]
13 LT
14 [JC 16]
15 [JMP 22]
16 [OUT '*']
17 [PFM 4]
18 [PFI 1]
19 ADD
20 [POP 4]
21 [JMP 11]
22 NL
23 [PFM 3]
24 [PFI 1]
25 SUB
26 [POP 3]
27 [JMP 4]
28 RET
29 [OUT '************************************************']
30 NL
31 [OUT 'ARTE ASCII EN PL/0.']
32 NL
33 [OUT 'DOS SECUENCIAS ENTERAS Y UN TRIANGULO RECTANGULO...']
```

```
34 NL
35 [OUT 'ANONIMO (S. XX)']
36 NL
37 [OUT '************************************************']
38 NL
39 [PFI 1]
40 [POP 0]
41 [PFM 0]
42 [PFI 20]
43 LTE
44 [JC 46]
45 [JMP 54]
46 [PFM 0]
47 OUT
48 [OUT ' ']
49 [PFM 0]
50 [PFI 1]
51 ADD
52 [POP 0]
53 [JMP 41]
54 [CAL 1]
55 [PFI 20]
56 NEG
57 [POP 1]
58 [PFI 0]
59 [POP 2]
60 [PFM 1]
61 [PFM 2]
62 LT
63 [JC 65]
64 [JMP 73]
65 [PFM 1]
66 OUT
67 [OUT ' ']
68 [PFM 1]
69 [PFI 1]
70 ADD
71 [POP 1]
72 [JMP 60]
73 NL
74 NL
75 HLT
```

```
PL/0> inter bien-08.pl0
************************************************
ARTE ASCII EN PL/0.
DOS SECUENCIAS ENTERAS Y UN TRIANGULO RECTANGULO...
ANONIMO (S. XX)
************************************************
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
***************
**************
*************
************
***********
**********
*********
********
*******
******
*****
****
***
**
*
-20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1
```

```
PL/0> escan bien-09.pl0
.
PL/0> virtu bien-09.pl0
0 HLT
PL/0> inter bien-09.pl0
PL/0>
```