

# GERAÇÃO DE CÓDIGO LINGUAGEM TINY PASCAL

Instrução	Hexa	Decimal	Argumento	Ação
LSP	4F	79	endereço	Stack Pointer←endereço
JMP	5A	90	endereço	Instruction P←endereço
LDI	44	68	inteiro	move inteiro para o topo da pilha
ADI	14	20	inteiro	adiciona inteiro ao topo da pilha
SUI	15	21	inteiro	subtrai inteiro do topo da pilha
MUI	16	22	inteiro	multiplica inteiro do topo da pilha
DVI	17	23	inteiro	divide inteiro do topo da pilha
LOD	40	64	endereço	carrega o conteúdo de end p/ pilha
STO	41	65	endereço	carrega o topo da pilha para end
OUT	58	88	-	pega o topo da pilha e mostra
IN	57	87	-	ler do teclado e coloca na pilha
JF	5C	92	endereço	pula para o endereço se falso
EQ (=)	20	32	-	verdade empilha \$FFFF falso \$0000
NE (not =)	21	33	-	verdade empilha \$FFFF falso \$0000
GT (>)	22	34	-	verdade empilha \$FFFF falso \$0000
GE (>=)	23	35	-	verdade empilha \$FFFF falso \$0000
LT (<)	24	36	-	verdade empilha \$FFFF falso \$0000
LE (<=)	25	37	-	verdade empilha \$FFFF falso \$0000
STOP	61	97	-	encerra a execução do programa

**EXEMPLO 01**

```
PROGRAM
VAR A:INTEGER;
BEGIN
    READ(A);
    IF A>10 THEN WRITE(A)
END
```

BYTE	INST / END	BYTE	INST/ END	BYTE	INST/END
0	LSP	12	LOD	24	00
1	00	13	06	25	OUT
2	10	14	00	26	STOP
3	JMP	15	LDI	27	
4	08	16	10	28	
5	00	17	00	29	
6	00	18	GT	30	
7	00	19	JF	31	
8	IN	20	26	32	
9	STO	21	00	33	
10	06	22	LOD	34	
11	00	23	06	35	

**EXEMPLO 02**

```
PROGRAM
VAR X1:INTEGER;
BEGIN
    WHILE X1<100 DO
        BEGIN
            X1 := X1+1;
            WRITE(X1)
        END
    END
END
```

BYTE	INST / END	BYTE	INST/ END	BYTE	INST/END
0	LSP	12	100	24	STO
1	00	13	00	25	06
2	10	14	LT	26	00
3	JMP	15	JF	27	LOD
4	08	16	34	28	06
5	00	17	00	29	00
6	00	18	LOD	30	OUT
7	00	19	06	31	JMP
8	LOD	20	00	32	08
9	06	21	ADI	33	00
10	00	22	01	34	STOP

11 | LDI  
-----

23 | 00  
-----

35

### EXEMPLO 03

```
PROGRAM
VAR X1:INTEGER;
BEGIN
  REPEAT
    BEGIN
      X1:=X1+1;
      WRITE(X1)
    END
  UNTIL X1 < 100
END
```

BYTE	INST / END
0	LSP
1	00
2	10
3	JMP
4	08
5	00
6	00
7	00
8	LOD
9	06
10	00
11	ADI

BYTE	INST/ END
12	01
13	00
14	STO
15	06
16	00
17	LOD
18	06
19	00
20	OUT
21	LOD
22	06
23	00

BYTE	INST/END
24	LDI
25	100
26	00
27	LT
28	JF
29	08
30	00
31	STOP
32	
33	
34	
35	

### EXEMPLO 04

```
PROGRAM
VAR
  NUM : INTEGER;
  MAX : INTEGER;
  MIN : INTEGER;
BEGIN
  READ(NUM) ;
  MAX := NUM;
  MIN := NUM;
  WHILE NUM <> 999 DO
    BEGIN
      IF NUM > MAX THEN MAX:=NUM;
      IF NUM < MIN THEN MIN:=NUM;
      READ(NUM)
    END;
    WRITE(MAX) ;
    WRITE(MIN)
  END
```

BYTE	INST / END
0	LSP
1	00
2	10

BYTE	INST/ END
12	IN
13	STO
14	06

BYTE	INST/END
24	00
25	STO
26	10

3		JMP
4		12
5		00
6		00
7		00
8		00
9		00
10		00
11		00

15		00
16		LOD
17		06
18		00
19		STO
20		08
21		00
22		LOD
23		06

27		00
28		LOD
29		06
30		00
31		LDI
32		99
33		09
34		NE
35		JF

BYTE		INST / END
36		77
37		00
38		LOD
39		06
40		00
41		LOD
42		08
43		00
44		GT (>)
45		JF
46		54
47		00

BYTE		INST/ END
48		LOD
49		06
50		00
51		STO
52		08
53		00
54		LOD
55		06
56		00
57		LOD
58		10
59		00

BYTE		INST/END
60		LT (<)
61		JF
62		70
63		00
64		LOD
65		06
66		00
67		STO
68		10
69		00
70		IN
71		STO

BYTE		INST / END
72		06
73		00
74		JMP
75		28
76		00
77		LOD

BYTE		INST/ END
78		08
79		00
80		OUT
81		LOD
82		10
83		00

BYTE		INST/END
84		OUT
85		STOP