Report #1

Submission	00682842_Entrega1_al021
File	project.lisp [Download]
Received	Sun Nov 08 22:38:35 WET 2015
Analyzed	Sun Nov 08 22:38:35 WET 2015 (0:00:00)
Team	al021 Login: `al021` Group: AL [AL]
Language	Lisp
Problem	A: Entrega
Compilation	

Test	Expected	Obtained	Differences
test01 T=0.004 sec E=0 sec M=64 Kb	;;␣Teste␣1\n ;;;␣Testes␣Tipo␣Accao\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n	;;␣Teste␣1\n ;;;␣Testes␣Tipo␣Accao\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n	;;␣Teste␣1\n ;;;␣Testes␣Tipo␣Accao\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n
test02 T=0.008 sec E=0 sec M=64 Kb	;;;␣Teste␣2\n ;;;␣Testes␣tipo␣tabuleiro␣1\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣0\n 0\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣2\n ;;;␣Testes␣tipo␣tabuleiro␣1\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣0\n 0\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣2\n ;;;␣Testes␣tipo␣tabuleiro␣1\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣0\n 0\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n
test03 T=0.008 sec E=0 sec M=64 Kb	;;;␣Teste␣3\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣3\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣3\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n
test04 T=0.008 sec E=0 sec M=64 Kb	;;;␣Teste␣4\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣6\n 6\n	;;;␣Teste␣4\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣6\n 6\n	;;;␣Teste␣4\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣6\n 6\n
test05 T=0.000 sec E=1 sec M=64 Kb	;;;␣Teste␣5\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n	;;;␣Teste␣5\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n	;;;␣Teste␣5\n ;;;␣Testes␣tipo␣tabuleiro␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n

	IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣3\n 3\n ;;deve␣retornar␣4\n 4\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣3\n 3\n	IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣3\n 3\n ;;deve␣retornar␣4\n 4\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣3\n 3\n	IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣3\n 3\n ;;deve␣retornar␣4\n 4\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣3\n 3\n
test06 T=0.008 sec E=0 sec M=64 Kb	;;;␣Teste␣6\n ;;;␣Testes␣tipo␣tabuleiro␣4\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣6\n ;;;␣Testes␣tipo␣tabuleiro␣4\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣6\n ;;;␣Testes␣tipo␣tabuleiro␣4\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n
test07 T=0.012 sec E=0 sec M=64 Kb	;;;␣Teste␣7\n ;;;␣Testes␣tipo␣estado␣1\n ;;retorna␣"ESTADO"\n ESTADO\n ;;retorna␣50\n 50\n ;;retorna␣'(I␣J)\n (I␣J)\n ;;retorna␣'(Z␣Z␣Z)\n (Z␣Z␣Z)\n ;;retorna␣99\n 99\n	;;;␣Teste␣7\n ;;;␣Testes␣tipo␣estado␣1\n ;;retorna␣"ESTADO"\n ESTADO\n ;;retorna␣50\n 50\n ;;retorna␣'(I␣J)\n (I␣J)\n ;;retorna␣'(Z␣Z␣Z)\n (Z␣Z␣Z)\n ;;retorna␣99\n 99\n	;;;␣Teste␣7\n ;;;␣Testes␣tipo␣estado␣1\n ;;retorna␣"ESTADO"\n ESTADO\n ;;retorna␣50\n 50\n ;;retorna␣'(I␣J)\n (I␣J)\n ;;retorna␣'(Z␣Z␣Z)\n (Z␣Z␣Z)\n ;;retorna␣99\n 99\n
test08 T=0.016 sec E=0 sec M=64 Kb	;;;␣Teste␣8\n ;;;␣Testes␣tipo␣estado␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;retorna␣50\n 50\n ;;retorna␣'(I␣J)\n (I␣J)\n ;;deve␣retornar␣NIL\n NIL\n ;;retorna␣'(Z␣Z␣Z)\n (Z␣Z␣Z)\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣8\n ;;;␣Testes␣tipo␣estado␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;retorna␣50\n 50\n ;;retorna␣'(I␣J)\n (I␣J)\n ;;deve␣retornar␣NIL\n NIL\n ;;retorna␣'(Z␣Z␣Z)\n (Z␣Z␣Z)\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣8\n ;;;␣Testes␣tipo␣estado␣2\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;retorna␣50\n 50\n ;;retorna␣'(I␣J)\n (I␣J)\n ;;deve␣retornar␣NIL\n NIL\n ;;retorna␣'(Z␣Z␣Z)\n (Z␣Z␣Z)\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n
test09 T=0.016 sec E=0 sec M=64 Kb	;;;␣Teste␣9\n ;;;␣Testes␣tipo␣estado␣3\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣9\n ;;;␣Testes␣tipo␣estado␣3\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣9\n ;;;␣Testes␣tipo␣estado␣3\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n
test10 T=0.020 sec E=0 sec M=64 Kb	;;;␣Teste␣10\n ;;;␣Testes␣tipo␣estado␣4\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣10\n ;;;␣Testes␣tipo␣estado␣4\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣10\n ;;;␣Testes␣tipo␣estado␣4\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣NIL\n NIL\n

	;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n	;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n	;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n
test11 T=0.016 sec E=0 sec M=64 Kb	;;;␣Teste␣11\n ;;;␣Testes␣tipo␣problema␣1\n ;;retorna␣"PROBLEMA"\n PROBLEMA\n ;;retorna␣T\n T\n ;;retorna␣1\n 1\n ;;retorna␣-1\n -1\n ;;retorna␣1\n 1\n ;;retorna␣-1\n -1\n	;;;␣Teste␣11\n ;;;␣Testes␣tipo␣problema␣1\n ;;retorna␣"PROBLEMA"\n PROBLEMA\n ;;retorna␣T\n T\n ;;retorna␣1\n 1\n ;;retorna␣-1\n -1\n ;;retorna␣1\n 1\n ;;retorna␣-1\n -1\n	;;;␣Teste␣11\n ;;;␣Testes␣tipo␣problema␣1\n ;;retorna␣"PROBLEMA"\n PROBLEMA\n ;;retorna␣T\n T\n ;;retorna␣1\n 1\n ;;retorna␣-1\n -1\n ;;retorna␣1\n 1\n ;;retorna␣-1\n -1\n
test12 T=0.008 sec E=0 sec M=64 Kb	;;;␣Teste␣12\n ;;;␣Testes␣fn␣solucao\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣12\n ;;;␣Testes␣fn␣solucao\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n	;;;␣Teste␣12\n ;;;␣Testes␣fn␣solucao\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n
test13 T=0.004 sec E=0 sec M=64 Kb	;;;␣Teste␣13\n ;;;␣Testes␣fn␣accoes\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣13\n ;;;␣Testes␣fn␣accoes\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n	;;;␣Teste␣13\n ;;;␣Testes␣fn␣accoes\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n ;;deve␣retornar␣T\n T\n
test14 T=0.012 sec E=0 sec M=64 Kb	;;;␣Teste␣14\n ;;;␣Testes␣fn␣resultado\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣'(I␣Z␣Z␣Z)\n (I␣Z␣Z␣Z)\n ;;deve␣retornar␣'(J)\n (J)\n ;;deve␣retornar␣50\n 50\n	;;;␣Teste␣14\n ;;;␣Testes␣fn␣resultado\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣'(I␣Z␣Z␣Z)\n (I␣Z␣Z␣Z)\n ;;deve␣retornar␣'(J)\n (J)\n ;;deve␣retornar␣50\n 50\n	;;;␣Teste␣14\n ;;;␣Testes␣fn␣resultado\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣NIL\n NIL\n ;;deve␣retornar␣'(I␣Z␣Z␣Z)\n (I␣Z␣Z␣Z)\n ;;deve␣retornar␣'(J)\n (J)\n ;;deve␣retornar␣50\n 50\n
test15 T=0.012 sec E=0 sec M=64 Kb	;;;␣Teste␣15\n ;;;␣Testes␣fn␣resultado\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n	;;;␣Teste␣15\n ;;;␣Testes␣fn␣resultado\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n	;;;␣Teste␣15\n ;;;␣Testes␣fn␣resultado\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣T\n

	T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣300\n 300\n ;;deve␣retornar␣NIL\n NIL\n	T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣300\n 300\n ;;deve␣retornar␣NIL\n NIL\n	T\n ;;deve␣retornar␣IGNORE\n IGNORE\n ;;deve␣retornar␣300\n 300\n ;;deve␣retornar␣NIL\n NIL\n
test16 T=0.012 sec E=0 sec M=64 Kb	;;;␣Teste␣16\n ;;;␣Testes␣fn␣qualidade\n ;;deve␣retornar␣-50\n -50\n	;;;␣Teste␣16\n ;;;␣Testes␣fn␣qualidade\n ;;deve␣retornar␣-50\n -50\n	;;;␣Teste␣16\n ;;;␣Testes␣fn␣qualidade\n ;;deve␣retornar␣-50\n -50\n
test17 T=0.004 sec E=0 sec M=64 Kb	;;;␣Teste␣17\n ;;;␣Testes␣fn␣custo-oportunidade\n ;;deve␣retornar␣700\n 700\n ;;deve␣retornar␣400\n 400\n ;;deve␣retornar␣400\n 400\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣850\n 850\n	;;;␣Teste␣17\n ;;;␣Testes␣fn␣custo-oportunidade\n ;;deve␣retornar␣700\n 700\n ;;deve␣retornar␣400\n 400\n ;;deve␣retornar␣400\n 400\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣850\n 850\n	;;;␣Teste␣17\n ;;;␣Testes␣fn␣custo-oportunidade\n ;;deve␣retornar␣700\n 700\n ;;deve␣retornar␣400\n 400\n ;;deve␣retornar␣400\n 400\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣200\n 200\n ;;deve␣retornar␣850\n 850\n

CPU	0.020 (1) sec
Memory	64 kbytes
Classification	Accepted
Mark	17
Observations	
Feedback	{17 tests with **Accepted**}
Code	

```
;;; GRUPO: 21 || ALUNOS: Henrique Lourenco - 77459 / Jose Touret - 78215 / Pedro Cruz - 78579

(load "utils.fas")

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;-------------------------------- Accao ---------------------------------------
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;CRIA-ACCAO
;;;construtor recebe um inteiro <c> correspondente a posicao da coluna mais a
;;;esquerda a partir da qual a peca vai ser colocada, e um array <peca> com a
;;;configuracao da peca a colocar, devolvendo uma nova accao
(defun cria-accao (c peca)
  (cons c peca)
  )

;;;ACCAO-COLUNA
;;;selector devolve um inteiro correspondente a coluna mais a esquerda a partir
;;;da qual a peca vai ser colocada
(defun accao-coluna (accao)
  (car accao)
  )

;;;ACCAO-PECA
;;;seletor devolve o array com a configuracao geometrica exacta com que vai
;;;ser colocada
(defun accao-peca (accao)
  (cdr accao)
  )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;----------------------------- Tabuleiro --------------------------------------
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;CRIA-TABULEIRO
;;;construtor que nao recebe qualquer argumento e devolve um novo tabuleiro vazio
(defun cria-tabuleiro ()
  (make-array '(18 10))
  )

;;;COPIA-TABULEIRO
;;;construtor que recebe um <tabuleiro> e devolve um novo tabuleiro com o mesmo
;;;conteudo do tabuleiro recebido
(defun copia-tabuleiro (tabuleiro)
  (let ((new-tabuleiro (cria-tabuleiro))
        (c (1- (cadr (array-dimensions tabuleiro))))
        (l (1- (car (array-dimensions tabuleiro))))
        )
```

```
    (dotimes (ic c)
      (dotimes (il l)
        (setf (aref new-tabuleiro il ic) (aref tabuleiro il ic))
        )
      )
    new-tabuleiro
    )
  )

;;;TABULEIRO-PREENCHIDO-P
;;;seletor que recebe um <tabuleiro>, um inteiro <l> que equivale ao numero da linha
;;;e um inteiro <c> que equivale ao numero da coluna e devolve o valor logico verdade
;;;se essa posicao estiver preenchida, falso caso contrario
(defun tabuleiro-preenchido-p (tabuleiro l c)
  (aref tabuleiro l c)
  )

;;;TABULEIRO-ALTURA-COLUNA
;;;seletor recebe um <tabuleiro>, um inteiro <c> correspondete ao numero de uma coluna
;;;e devolve a altura da coluna, ou seja, a posicao mais alta preenchida dessa coluna
(defun tabuleiro-altura-coluna (tabuleiro c)
  (let ((l (car (array-dimensions tabuleiro))))
    (dotimes (i l)
      (if (tabuleiro-preenchido-p tabuleiro  (1- (- l  i)) c)
          (return-from tabuleiro-altura-coluna  (- l  i))
        )
      )
    0
    )
  )

;;;TABULEIRO-LINHA-COMPLETA-P
;;;reconhecedor recebe um <tabuleiro>, um inteiro que equivale ao numero de uma
;;;linha <l> e devolve o valor logico verdade se todas as posicoes da linha
;;;recebida estiverem preenchidas, e falso caso contrario
(defun tabuleiro-linha-completa-p (tabuleiro l)
  (let ((c (1- (cadr (array-dimensions tabuleiro)))))
    (dotimes (i c)
      (cond ((not (tabuleiro-preenchido-p tabuleiro l  i)) (return-from tabuleiro-linha-completa-p nil)))
      )
    )
  t
  )

;;;TABULEIRO-PREENCHE!
;;;modificador recebe um <tabuleiro>, um inteiro <l> que equivale ao numero
;;;da linha e um inteiro <c> que equivale ao numero da coluna e altera o
;;;tabuleiro recebido para a posicao correspondente a linha e coluna passar a
;;;estar preenchido
(defun tabuleiro-preenche! (tabuleiro l c)
  (if (and (< l (car (array-dimensions tabuleiro))) (>= l 0))
      (if (and (< c (cadr (array-dimensions tabuleiro))) (>= c 0))
          (setf (aref tabuleiro l c) T)
        )
    )
  )

;;;TABULEIRO-REMOVE-LINHA!
;;;modificador que recebe um <tabuleiro>, um inteiro <l> correspondente ao
;;;numero da linha, e altera o tabuleiro recebido removendo essa linha do
;;;tabuleiro, e fazendo com que as linhas por cima da linha removida descam
;;;uma linha
(defun tabuleiro-remove-linha! (tabuleiro l)
  (if (and (<= l (car (array-dimensions tabuleiro))) (>= l 0))
      (let ((c (cadr (array-dimensions tabuleiro))))
        (dotimes (i c)
          (dotimes (j (1- (- (car (array-dimensions tabuleiro)) l)))
            (setf (aref tabuleiro (+ j l) i) (aref tabuleiro (1+ (+ j l)) i))
            )
          (setf (aref tabuleiro (1- (car (array-dimensions tabuleiro))) i) NIL)
          )
        )
    )
  )

;;;TABULEIRO-TOPO-PREENCHIDO-P
;;;reconhecedor recebe um <tabuleiro>, e devolve o valor logico verdade se
;;;existir alguma posicao na linha do topo do tabuleiro que esteja preenchida,
;;;e falso caso contrario
(defun tabuleiro-topo-preenchido-p (tabuleiro)
  (let ((c (1- (cadr (array-dimensions tabuleiro)))))
    (dotimes (i c)
      (cond ((tabuleiro-preenchido-p tabuleiro 17  i) (return-from tabuleiro-topo-preenchido-p t)))
      )
    )
  nil
  )

;;;TABULEIROS-IGUAIS-P
;;;teste recebe dois tabuleiros <t1> e <t2>, e devolve o valor logico verdade se
;;;os dois tabuleiros forem iguais, e falso caso contrario
(defun tabuleiros-iguais-p (t1 t2)
  (let ((l (1- (car (array-dimensions t1))))
        (c (1- (cadr (array-dimensions t1))))
        )
    (if (or (not (equalp l (1- (car (array-dimensions t2))))) (not (equalp c (1- (cadr (array-dimensions t1))))))
        (return-from tabuleiros-iguais-p nil)
```

```
      )
    (dotimes (i l)
      (dotimes (j c)
        (if (not (equalp (aref t1 i j) (aref t2 i j)))
            (return-from tabuleiros-iguais-p nil)
          )
        )
      )
    t
    )
  )

;;;TABULEIRO->ARRAY
;;;transformador de saida recebe um <tabuleiro> e devolve um novo array, que para
;;;cada linha e coluna devera conter o valor logico correspondente a cada posicao
;;;do tabuleiro
(defun tabuleiro->array (tabuleiro)
  tabuleiro
  )

;;;ARRAY->TABULEIRO
;;;transformador de entrada recebe um <array> cujas posicoes logicas tem o valor
;;;logico T ou Nil, e constroi um novo tabuleiro com o conteudo do array recebido
(defun array->tabuleiro (array)
  array
  )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;------------------------------ Estado ------------------------------------
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;ESTADO - representa o estado de um jogo de tetris
;;;<pontos> - numero de pontos conseguidos ate ao momento
;;;<pecas-por-colocar> - lista contendo as pecas por colocar, por ordem de colocacao
;;;<pecas-colocadas> - lista com as pecas ja colocadas no tabuleiro
;;;<tabuleiro> - tabuleiro com as posicoes atualmente preenchidas do jogo
(defstruct estado pontos pecas-por-colocar pecas-colocadas tabuleiro)

;;;COPIA-ESTADO
;;;construtor que recebe um <estado> e devolve um novo cujo conteudo deve ser
;;;copidado a partir do estado original
(defun copia-estado (estado)
  (make-estado :pontos (estado-pontos estado)
               :pecas-por-colocar (copy-list (estado-pecas-por-colocar estado))
               :pecas-colocadas (copy-list (estado-pecas-colocadas estado))
               :tabuleiro (copia-tabuleiro (estado-tabuleiro estado))
               )
  )

;;;ESTADOS-IGUAIS-P
;;;teste que recebe dois estados <estado1> e <estado2> , devolvendo o valor logico
;;;verdade se os dois estados forem iguais e falso caso contrario
(defun estados-iguais-p (estado1 estado2)
  (equalp estado1 estado2)
  )

;;;ESTADO-FINAL-P
;;;reconhecedor recebe um <estado> e devolve o valor logico verdade se corresponder a
;;;um estado final onde o jogador ja nao possa fazer mais jogadas e falso caso
;;;contrario
(defun estado-final-p (estado)
  (if (or (not (car (estado-pecas-por-colocar estado))) (tabuleiro-topo-preenchido-p (estado-tabuleiro estado)))
      T
    NIL
    )
  )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;---------------------------- Problema ------------------------------------
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;PROBLEMA - representa um problema generico de procura
;;;<estado-inicial> - contem o estado inicial do problema de procura
;;;<solucao> - funcao que verifica se um estado e solucao de um problema de procura
;;;<accoes> - funcao que devolve uma lista com todas as accoes possiveis para um
;;;         estado recebido
;;;<resultado> - funcao que devolve o estado sucessor que resulta de executar a accao
;;;         recebida no estado recebido
;;;<custo> - caminho - funcao que devolve o custo do caminho desde o estado incicial
;;;        ate um estado recebido
(defstruct problema estado-inicial solucao accoes resultado custo-caminho)

;;;SOLUCAO
;;;funcao recebe um <estado> e devolve o valor logico verdade se o estado recebido
;;;corresponder a uma solucao, e falso caso contrario
(defun solucao (estado)
  (if (and (not (car (estado-pecas-por-colocar estado))) (not (tabuleiro-topo-preenchido-p (estado-tabuleiro estado))))
      t
    )
  )

;;;ACCOES
;;;funcao recebe um <estado> e devolve uma lista de accoes correspondendo a todas as
```

```
;;;accoes validas que podem ser feitas com a proxima peca a ser colocada
(defun accoes (estado)
  (let ((c (cadr (array-dimensions (estado-tabuleiro estado))))
        (validos nil)
        (resposta ()))
    (cond ((equalp 'i (car (estado-pecas-por-colocar estado))) (setf validos (list peca-i0 peca-i1)))
          ((equalp 'j (car (estado-pecas-por-colocar estado))) (setf validos (list peca-j0 peca-j1 peca-j2 peca-j3)))
          ((equalp 'l (car (estado-pecas-por-colocar estado))) (setf validos (list peca-l0 peca-l1 peca-l2 peca-l3)))
          ((equalp 'o (car (estado-pecas-por-colocar estado))) (setf validos (list peca-o0)))
          ((equalp 's (car (estado-pecas-por-colocar estado))) (setf validos (list peca-s0 peca-s1)))
          ((equalp 'z (car (estado-pecas-por-colocar estado))) (setf validos (list peca-z0 peca-z1)))
          ((equalp 't (car (estado-pecas-por-colocar estado))) (setf validos (list peca-t0 peca-t1 peca-t2 peca-t3)))
          )
    (dolist(peca validos)
      (dotimes (i (- c (1- (cadr (array-dimensions peca)))))
        (setf resposta (append resposta (list (cria-accao i peca))))
        )
      )
    resposta
    )
  )

;;;RESULTADO
;;;funcao recebe um <estado> e uma <accao>, e devolve um novo estado que resulta de
;;;aplicar a accao recebida ao estado original
(defun resultado (estado accao)
  (let ((estado2 (copia-estado estado))
        (altura 0)
        (l (car (array-dimensions (cdr accao))))
        (c (cadr (array-dimensions (cdr accao))))
        (linhas-removidas 0))
    (setf (estado-pecas-colocadas estado2) (append (list (car (estado-pecas-por-colocar estado2))) (estado-pecas-colocadas estado2)))
    (setf (estado-pecas-por-colocar estado2) (cdr (estado-pecas-por-colocar estado2)))
    (dotimes (i l)
      (if (< altura (tabuleiro-altura-coluna (estado-tabuleiro estado2) (+ (car accao) i)))
          (setf altura (tabuleiro-altura-coluna (estado-tabuleiro estado2) (+ (car accao) i)))
        )
      )
    (dotimes (i l)
      (dotimes (j c)
        (if (aref (cdr accao)  i j)
            (tabuleiro-preenche! (estado-tabuleiro estado2) (+ i altura) (+ j (car accao)))
          )
        )
      )
    (if (tabuleiro-topo-preenchido-p (estado-tabuleiro estado2))
        (return-from resultado estado2)
      (dotimes (i (car (array-dimensions (estado-tabuleiro estado2))))
        (if (tabuleiro-linha-completa-p (estado-tabuleiro estado2) i)
            (progn
              (tabuleiro-remove-linha! (estado-tabuleiro estado2) i)
              (setf linhas-removidas (1+ linhas-removidas))
              (setf i (1- i))
              )
          )
        )
      )
    (cond ((equalp linhas-removidas 1) (setf (estado-pontos estado2) (+ (estado-pontos estado2) 100)))
          ((equalp linhas-removidas 2) (setf (estado-pontos estado2) (+ (estado-pontos estado2) 300)))
          ((equalp linhas-removidas 3) (setf (estado-pontos estado2) (+ (estado-pontos estado2) 500)))
          ((equalp linhas-removidas 4) (setf (estado-pontos estado2) (+ (estado-pontos estado2) 800)))
          )
    estado2
    )
  )

;;;QUALIDADE
;;;funcao que recebe um <estado> e retorna um valor de qualidade que corresponde ao
;;;valor negativo dos pontos ganhos ate ao momento
(defun qualidade (estado)
  (* (estado-pontos estado) -1)
  )

;;;CUSTO-OPORTUNIDADE
;;;funcao que recebe um <estado> e devolve o custo de oportunidade de todas as accoes
;;;realizadas ate ao momento
(defun custo-oportunidade (estado)
  (let ((resposta 0))
    (dolist(peca (estado-pecas-colocadas estado))
      (cond ((equalp 'i (car (estado-pecas-colocadas estado))) (setf resposta (+ resposta 800)))
            ((equalp 'j (car (estado-pecas-colocadas estado))) (setf resposta (+ resposta 500)))
            ((equalp 'l (car (estado-pecas-colocadas estado))) (setf resposta (+ resposta 500)))
            ((equalp 'o (car (estado-pecas-colocadas estado))) (setf resposta (+ resposta 300)))
            ((equalp 's (car (estado-pecas-colocadas estado))) (setf resposta (+ resposta 300)))
            ((equalp 'z (car (estado-pecas-colocadas estado))) (setf resposta (+ resposta 300)))
            ((equalp 't (car (estado-pecas-colocadas estado))) (setf resposta (+ resposta 300)))
            )
      )
    (- resposta (estado-pontos estado))
    )
  )

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;------------------------------ FIM -------------------------------------------
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```