Código fuente en ADA	Código de 3 direcciones	Cuádruplas

procedure Factorial(x:in integer, fact:out integer)
is
  tmp : integer
begin
  -- Calcula x!
  if (x<0) then
    fact := 1;
  elseif (x=0) then
    fact := 1;
  else
    tmp := 1;
    while (x>0) loop
      tmp := tmp * x;
      x := x - 1;
    end loop;
    fact := tmp;
  end if;
end;
	Label L01
	tmp1 = x < 0
	if false tmp1 goto L02
	fact = 1
	goto L06
	
	Label L02
	tmp2 = x = 0
	if false tmp2 goto L03
	fact = 1
	goto L06
	
	Label L03
	tmp = 1
	
	Label L04
	tmp3 = x > 0
	if false tmp3 goto L05
	tmp4 = tmp * x
	tmp = tmp4
	tmp5 = x - 1
	x = tmp5
	goto L04
	
	Label L05
	fact = tmp
	
	Label L06
		(label,L01,-,-)
		(less_than,tmp1,x,0)
		(if_false,tmp1,L02,-)
		(assign,fact,1,-)
		(goto,L06,-,-)
		
		(label,L02,-,-)
		(is_equal,tmp2,x,0)
		(if_false,tmp2,L03,-)
		(assign,fact,1,-)
		(goto,L06,-,-)
		
		(label,L03,-,-)
		(assign,tmp,1,-)
		
		(label,L04,-,-)
		(greater_than,tmp3,x,0)
		(if_false,tmp3,L05,-)
		(MUL,tmp4,tmp,x)
		(assign,tmp,tmp4,x)
		(SUB,tmp5,x,1)
		(assign,x,tmp5,-)
		(goto,L04,-,-)
		
		(label,L05,-,-)
		(assign,fact,tmp,-)
		
		(label,L06,-,-)

Código fuente en C	Código de 3 direcciones	Tripletas
for (i = 1; i < 500 ; i++) {	i =1	(1)(assign, i, 1)
switch (a [i]) {		(2)(less_than, i, 500)
case 0:	label L1	(3)(if_false, (2), (29))
c [i] = a [i - 1] ;	temp1 = (i < 500)	(4)(=[ ], A, i)
b [i] = c [i] ;	if_false temp1 goto L2	(5)(is_equal, (4), 0)
break;	temp2 = A[i]	(6)(if_false, (5), (15))
case 1:	temp3 = (temp2 = 0)	(7)(SUB, i , 1)
c [i] = b [i] ;	if_false temp3 goto L4	(8)(=[ ], A, (7))
b [i] = 0 ;	temp4 = i - 1	(9)([ ]=, C, i)
break;	temp5 = A[temp4]	(10)(assign, (9), (8))
default:	C[i] = temp5	(11)(=[ ], C, i)
c [i] = a [i] ;	temp6 = C[i]	(12)([ ]=, B, i)
}	B[i] = temp6	(13)(assign, (12), (11))
}	goto L3	(14)(goto, (26), -)
		(15)(is_equal, (4) , 1)
	label L4	(16)(if_false, (15), (23))
	temp7 = (temp2 = 1)	(17)(=[ ], B, i)
	if_false temp7 goto L5	(18)([ ]=, C, i)
	temp8 = B[i]	(19)(assign, (18), (17))
	C[i] = temp8	(20)([ ]=, B, i)
	B[i] = 0	(21)(assign, (22), 0)
	goto L3	(22)(goto, (26), -)
		(23)(=[ ], A, i)
	label L5	(24)([ ]=, C, i)
	temp9 = A[i]	(25)(assign, (24), (23))
	C[i] = temp9	(26)(SUMA, i, 1)
		(27)(assign, i, (26))
	label L3	(28)(goto, (2), -)
	tmp10 = i + 1	(29) ...
	i = tmp10	
	goto L1	
	label L2	