```
X,Y : ARRAY[1..10,1..10] OF INTEGER
          .
          .
          .
FOR I := 1 TO 10 DO
   X[I,2*J-1] := Y[I,2*J]
```

(a)

(1)	:=	#1		I	{loop initialization}
(2)	JGT	I	#10	(20)	
(3)	-	I	#1	i_1	{subscript calculation for X}
(4)	*	i_1	#10	i_2	
(5)	*	#2	J	i_3	
(6)	-	i_3	#1	i_4	
(7)	-	i_4	#1	i_5	
(8)	+	i_2	i_5	i_6	
(9)	*	i_6	#3	i_7	
(10)	-	I	#1	i_8	{subscript calculation for Y}
(11)	*	i_8	#10	i_9	
(12)	*	#2	J	i_{10}	
(13)	-	i_{10}	#1	i_{11}	
(14)	+	i_9	i_{11}	i_{12}	
(15)	*	i_{12}	#3	i_{13}	
(16)	:=	Y[i_{13}]		X[i_7]	{assignment operation}
(17)	+	#1	I	i_{14}	{end of loop}
(18)	:=	i_{14}		I	
(19)	J			(2)	
(20)					{next statement}

(b)

(1)	:=	#1		I	{loop initialization}
(2)	JGT	I	#10	(16)	
(3)	-	I	#1	i_1	{subscript calculation for X}
(4)	*	i_1	#10	i_2	
(5)	*	#2	J	i_3	
(6)	-	i_3	#1	i_4	
(7)	-	i_4	#1	i_5	
(8)	+	i_2	i_5	i_6	
(9)	*	i_6	#3	i_7	
(10)	+	i_2	i_4	i_{12}	{subscript calculation for Y}
(11)	*	i_{12}	#3	i_{13}	
(12)	:=	Y[i_{13}]		X[i_7]	{assignment operation}
(13)	+	#1	I	i_{14}	{end of loop}
(14)	:=	i_{14}		I	
(15)	J			(2)	
(16)					{next statement}

(c)

Figure 5.27 Code optimization by elimination of common subexpressions and removal of loop invariants.