Functions for extracting information from i^{th} instruction of program number i.

Input of fuctions: i value i for i^{th} instruction.
y is the program #.

Recall that program # y is created as follows:

$[\#I_1, \#I_2, \ldots, \#I_n] - 1 = \#P = y$

Hence $(y+1)i$ is the value or number of the i^{th} instruction. That is, it is some $\langle a, \langle b, c \rangle \rangle$ for the i^{th} instruction

$Label(i,y) = \ell((y+1)i)$
This is the value "a" of any label in the set of labels. If no label it is 0.

$Var(i;y) = r(r(\ell(y+1)i)) + 1$
This is the index # of the variable of the instruction. Note that it was necessary to get $\langle b,c \rangle$ first, then "c" and then add 1 to that value "c"

$INSTR(i,y) = \ell(r(((y+1)i))$
This is the value of b. Note is can be 0, 1, 2, or higher if it is a label pointer for IF inst.

$LABEL'(i,y) = \ell(r((y+1)i)) \dot{-} 2$
The value of the actual label #, or 0 is it a substract inst.

A snapshot has 2 values. The index of the instruction be executed, say i. Also, the state of the computation or values for all variables. We originally used (i, σ) for a snapshot. We now use $\langle i, s \rangle$ for a snapshot. This "number" is using pairing Here i is the instruction #. However s is a number representing the state of the computation. Hence it is the #

$s = \prod_{i=1}^{n} p_i^{x_i}$ for all indices, n, of variables that have nonzero values.

We that create the 'pair" value $x = \langle i, s \rangle$

The next set of predicates relate to this x, and program y.

SKIP(x,y) : this predicate and other predicates
INCR(x,y)
DECR(x,y)
BRANCH(x,y) operate on the same x,y as above.

VAR(i(x),y) is: the index of the variable in the current instruction i.

PVAR() is: the prime # in s whose exponent in the current value of the variable.

r(x) is: the value $s = \prod_{i=1}^{n} p_i^{x_i}$

PVAR(i(x),y) | r(x) means the values of v_i divides s.

SKIP (x,y) : It is true if the instruction value b in $\langle a,\langle b,c\rangle\rangle$ = 0, or if if the value b = 2 and the index value of Label has value 0.

INCR (x,y) : need to do an add

DECR (x,y) : need to do a subtract & value of variable is = 1.

BRANCH (x, Y) : if ∃ a value of a label to branch to.