$P \rightarrow \{ offset = 0; \}\ D$

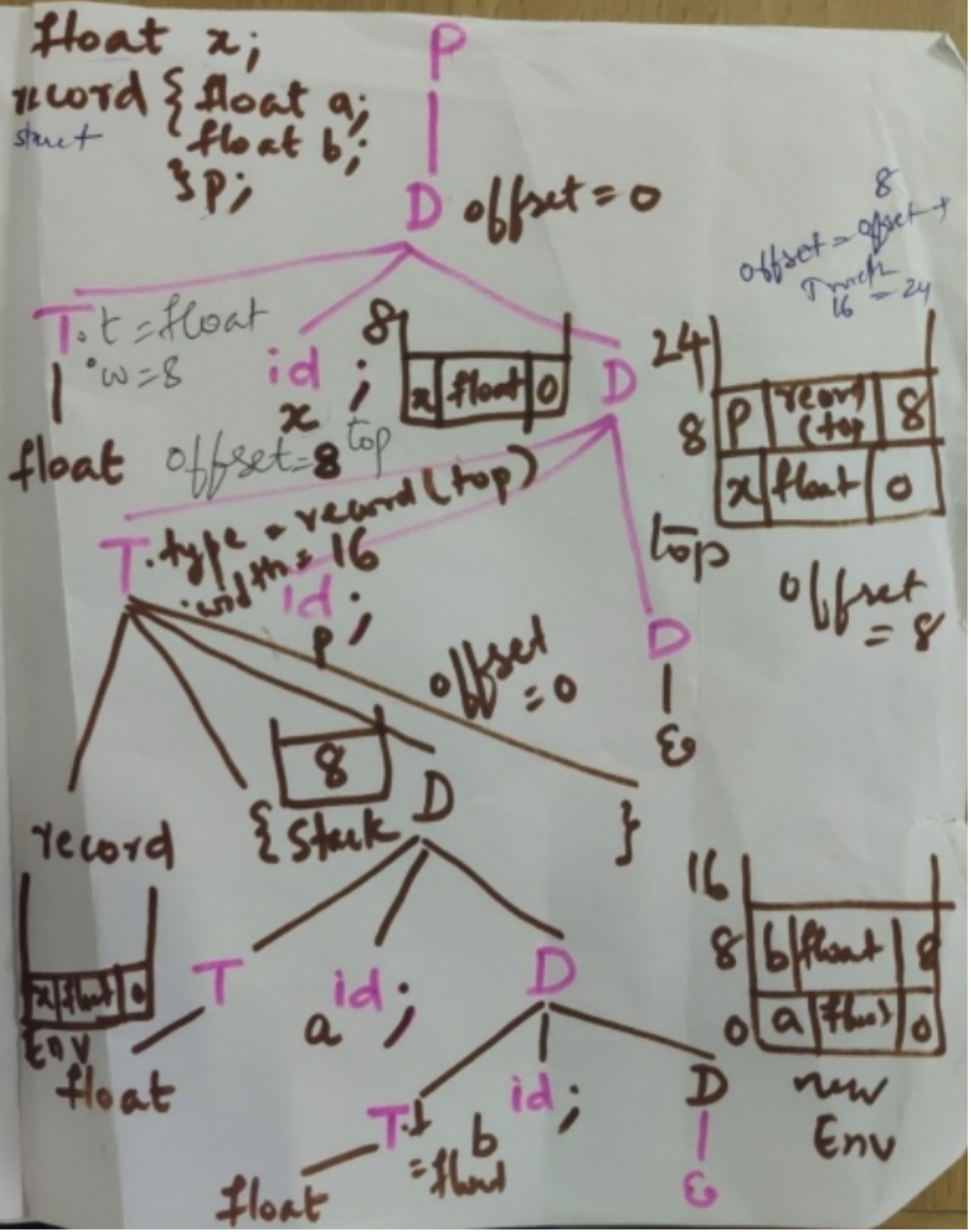
$D \rightarrow T\ id;\ \{ top.put($
$id.lexeme, T.type, offset);$
$offset = offset + T.width; \}$

D_1

$D \rightarrow \varepsilon$

$T \rightarrow record\ `\{'\ \{ Env.push(top);$
$top = new\ Env();$
$Stack.push(offset);$
$offset = 0; \}$

$D\ `\}'\ \{ T.type = record(top);$
$T.width = offset;$
$top = Env.pop();$
$offset = Stack.pop(); \}$



Computing type and their widths

int [5] [5];

int [5][5]

⇒

type expression

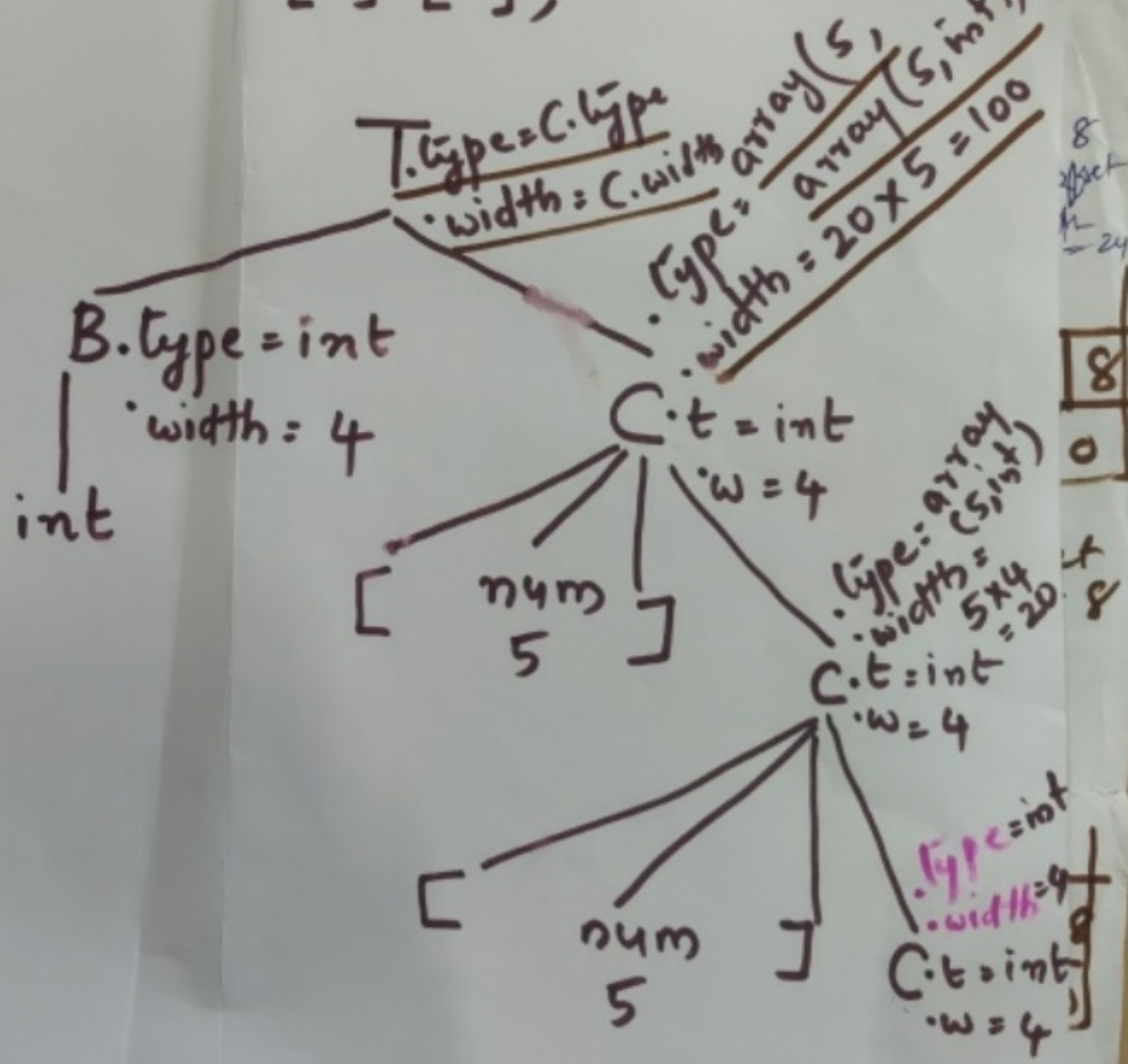
array(5, array(5, int))

Total storage units

= 5 × 5 × 4

(width of integer)

= 100



row-major

A

a_{00}	a_{01}	a_{02}	a_{03}	a_{10}	a_{11}	a_{12}	a_{13}	a_{20}	a_{21}	a_{22}	a_{23}
100	102	104	106	108	110	112	114	116	118	120	122

size = 2

2 bytes

~~pleas~~. n = columns.

add (A[2][1]) = [L_0 + $i * n + j$] * w   A[2][1]

(Base address → L_0)

= 100 + 9 * 2 = 118.

formula is

if index starts with 0

add (A[i][j]) = L_0 + [i * n + j] * w

index from 1 formula is

add (A[i][j]) = L_0 + [(i-1) * n + (j-1)] * w

m = rows

n = columns.

8

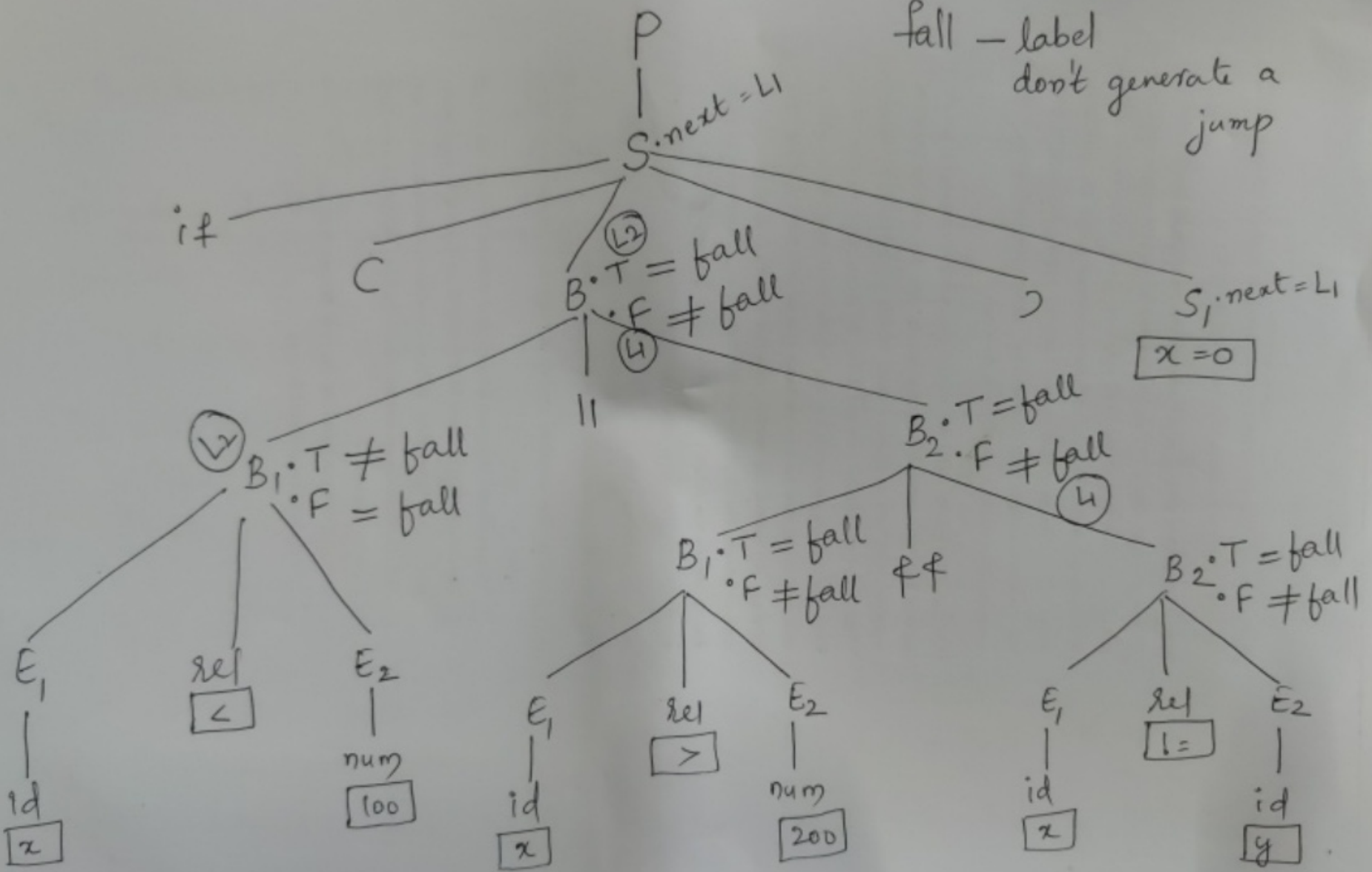
Offset

= 24

8 | 0

Offset = 8

if (x<100 || x>200 && x != y) x=0;



$B \rightarrow B_1 \;||\; B_2$

B.code = ~~B.true~~ ≠ fall

Yes ~~B.T ≠ fall then~~

B.code = B.true = fall

Yes, B.true = fall then ⇒

B_1.code ⇒

if x < 100 goto L2

B_2.code ⇒

iffalse x > 200 goto L1

iffalse x != y goto L1

B.code = B_1.code || B_2.code || label(B_1.true)

if x < 100 goto L2

iffalse x > 200 goto L1

iffalse x != y goto L1

L2 :

test = E_1.addr rel.op E_2.addr

① E_1.addr = x ; rel.op = <
E_2.addr = 100

B.T ≠ fall then

if x < 100 goto L2

② E.addr = x ; rel.op = >
E_2.addr = 200

B.F ≠ fall then

iffalse x > 200 goto L_1

③ E.addr = x ; rel.op = !=
E.addr = y

B.F ≠ fall then

iffalse x != y goto L_1

$S \rightarrow if\ (B)\ S_1$

B.code is compiled

S_1.code ⇒ `x = 0`

if x <100 goto L2

iffalse x > 200 goto L1

iffalse x != y goto L1

L2 : x = 0

L1 : — — —

$S.code = B.code \mid\mid S_1.code$

~~S.code = B.code || label (B.true) || S_1.code~~

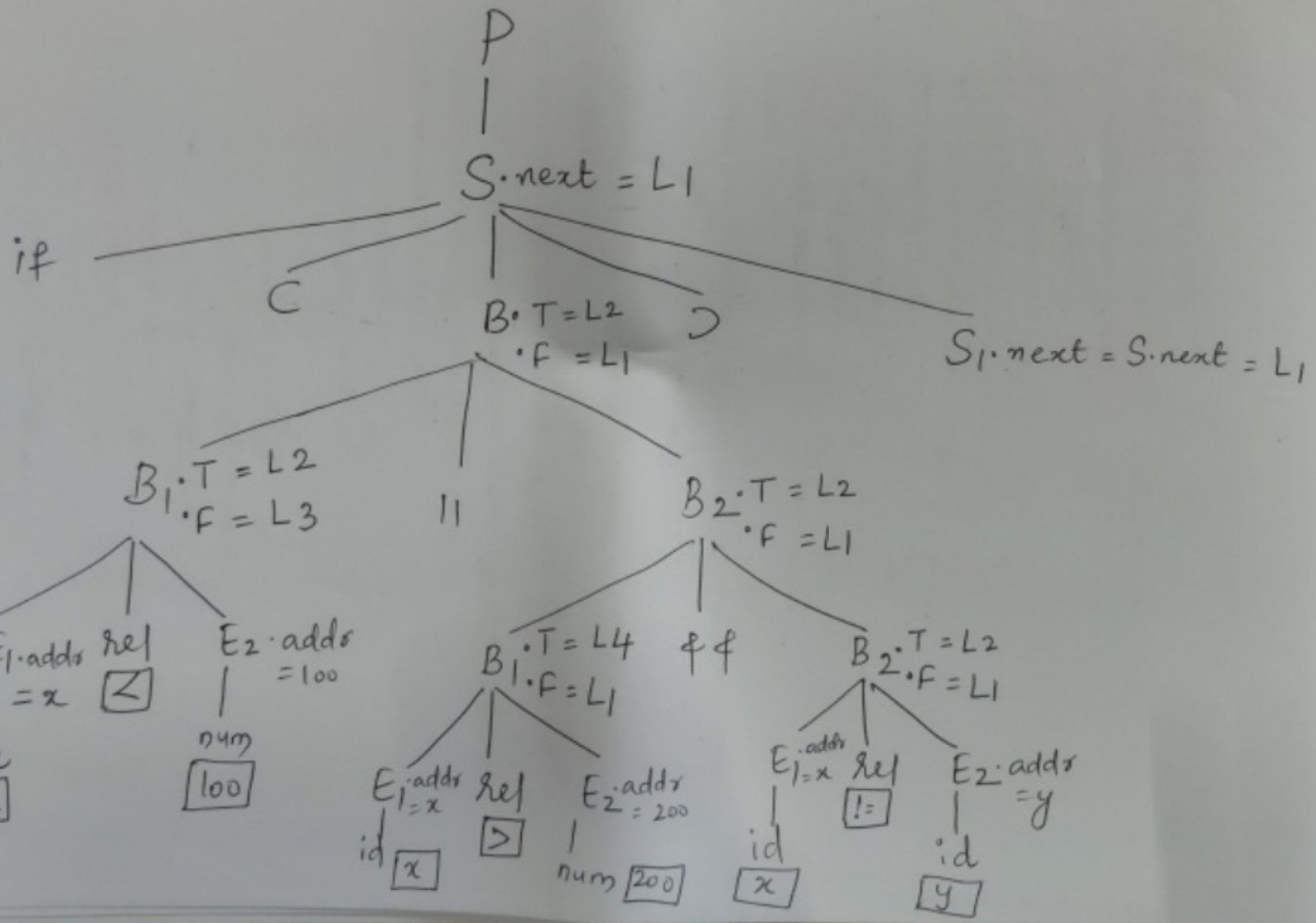
B.code
L2 : x = 0

$P \rightarrow S$

$P.code = S.code \mid\mid label(S.next)$

S.code
L1 : ____

if (x < 100 || x > 200 && x != y) x = 0;

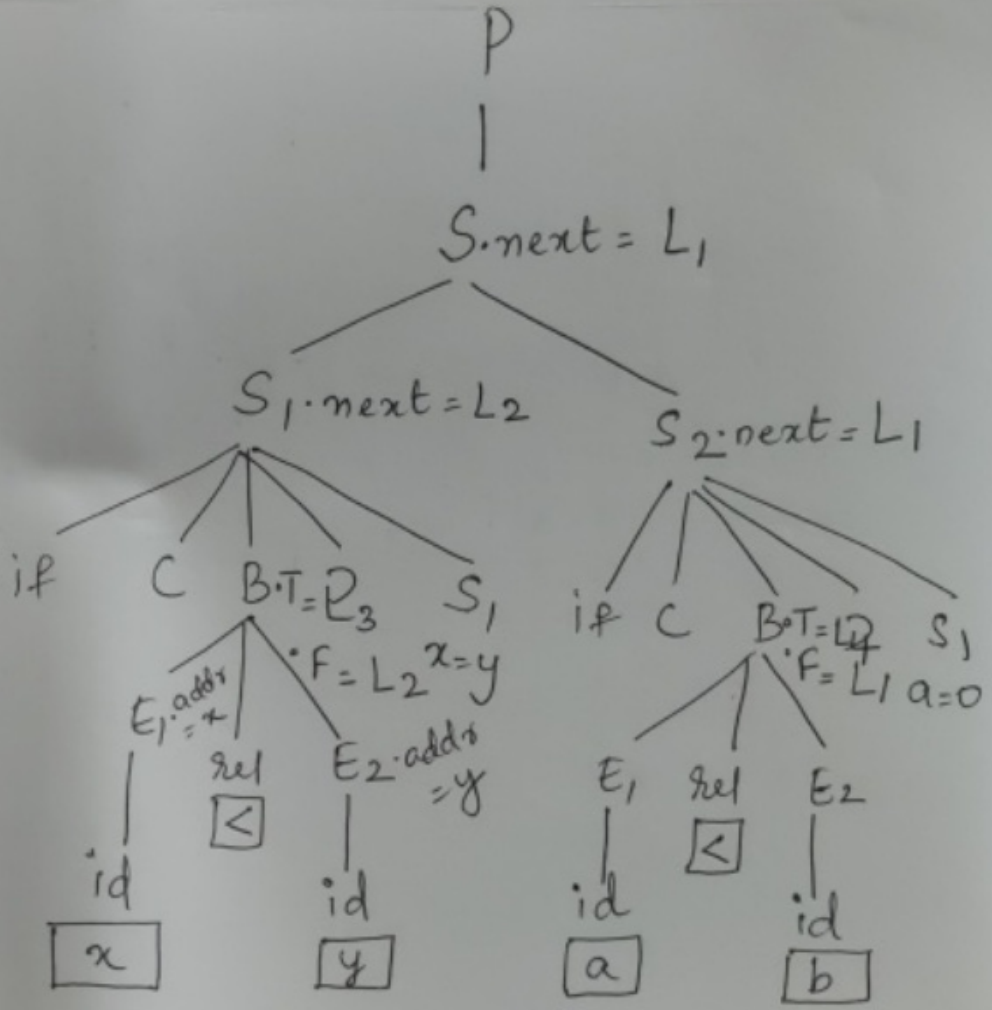


if (a==b && c==d || e==f) x==1;

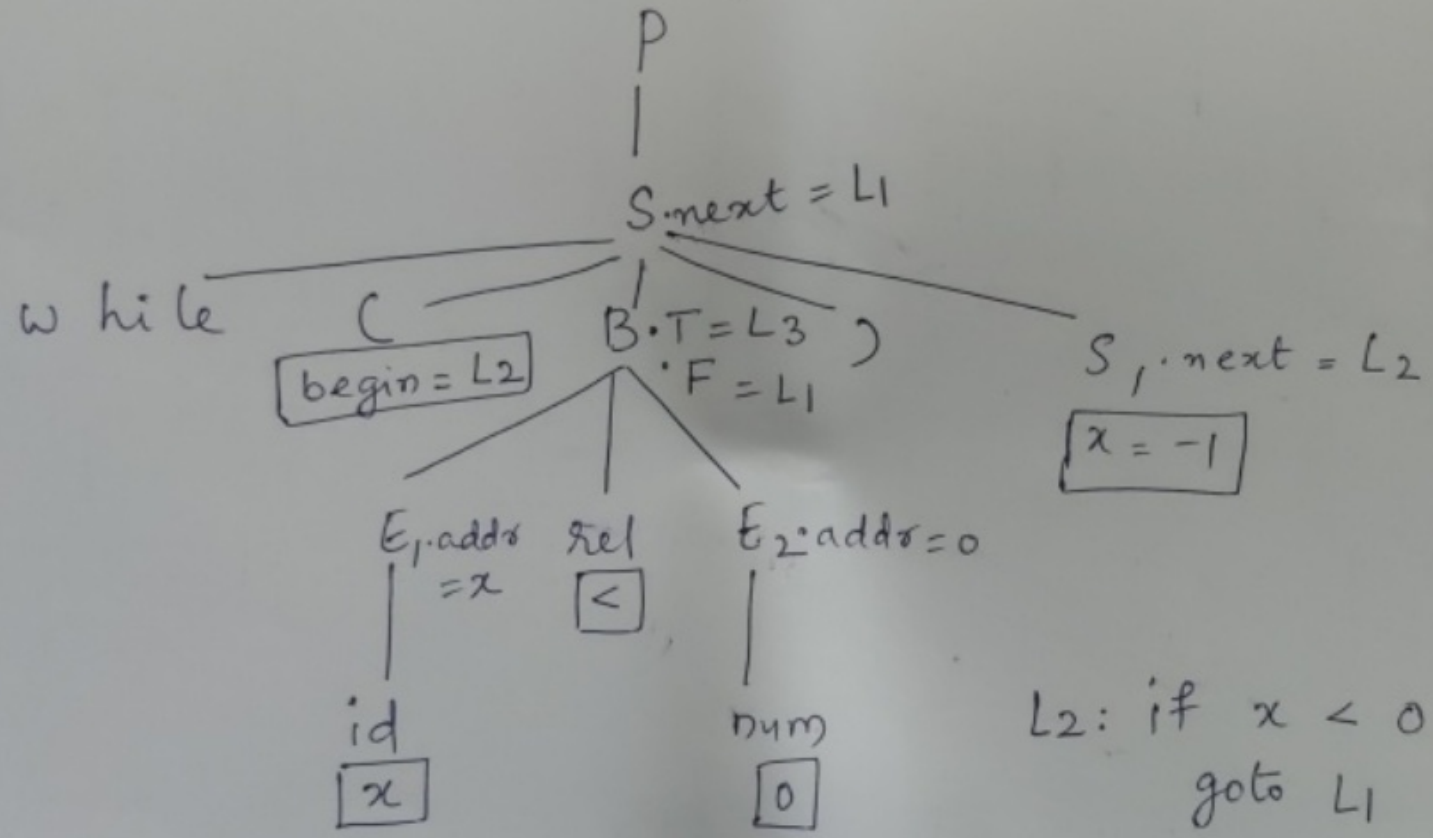
if (x<y) x=y;
if (a<b) a=0;

Three Address Code

if x<y goto L_3
goto L_2
L_3 : x=y
L_2 : if a<b goto L_4
goto L_1
L_4 : a=0
L_1 : — — —

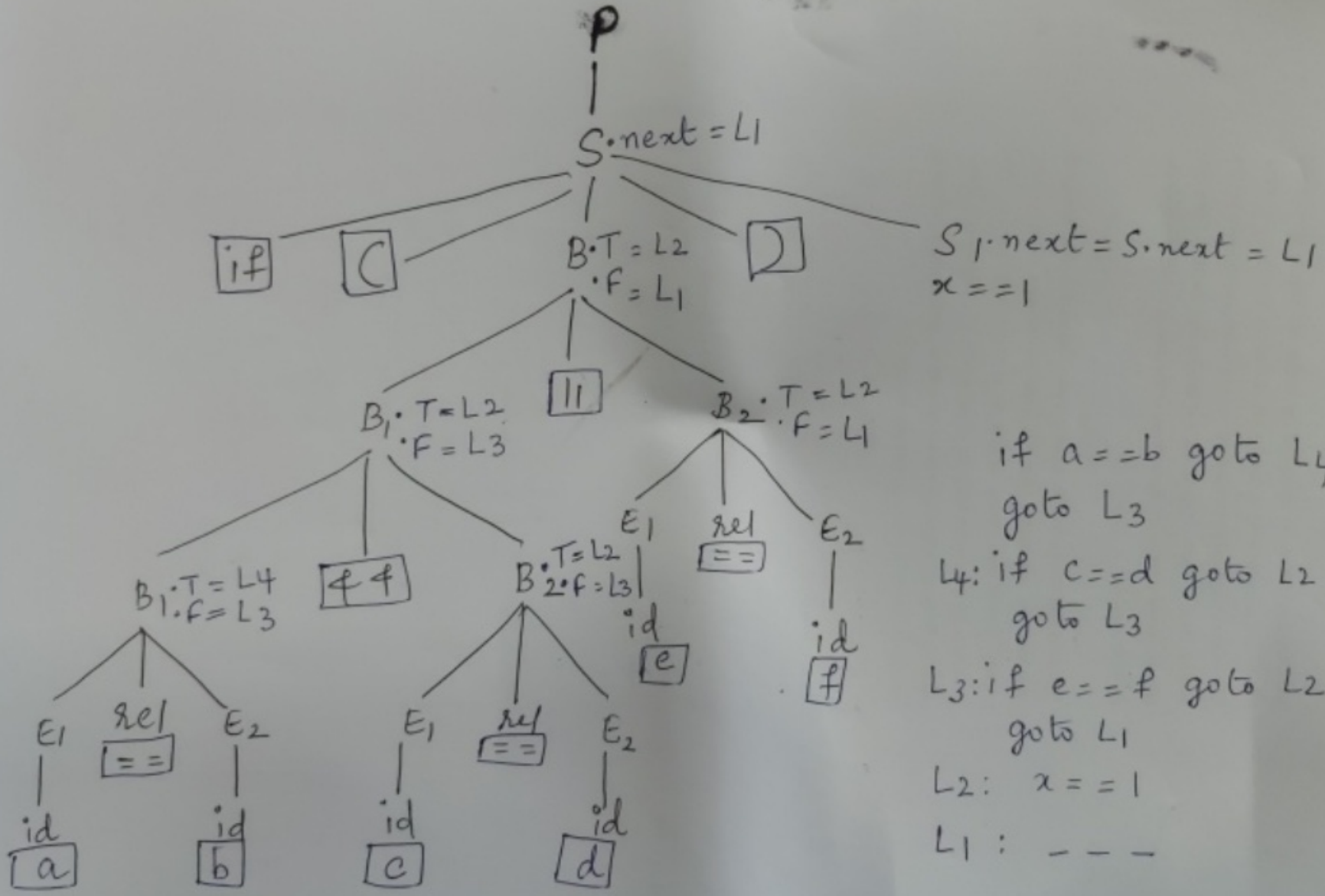


while (x < 0) x = -1 ;



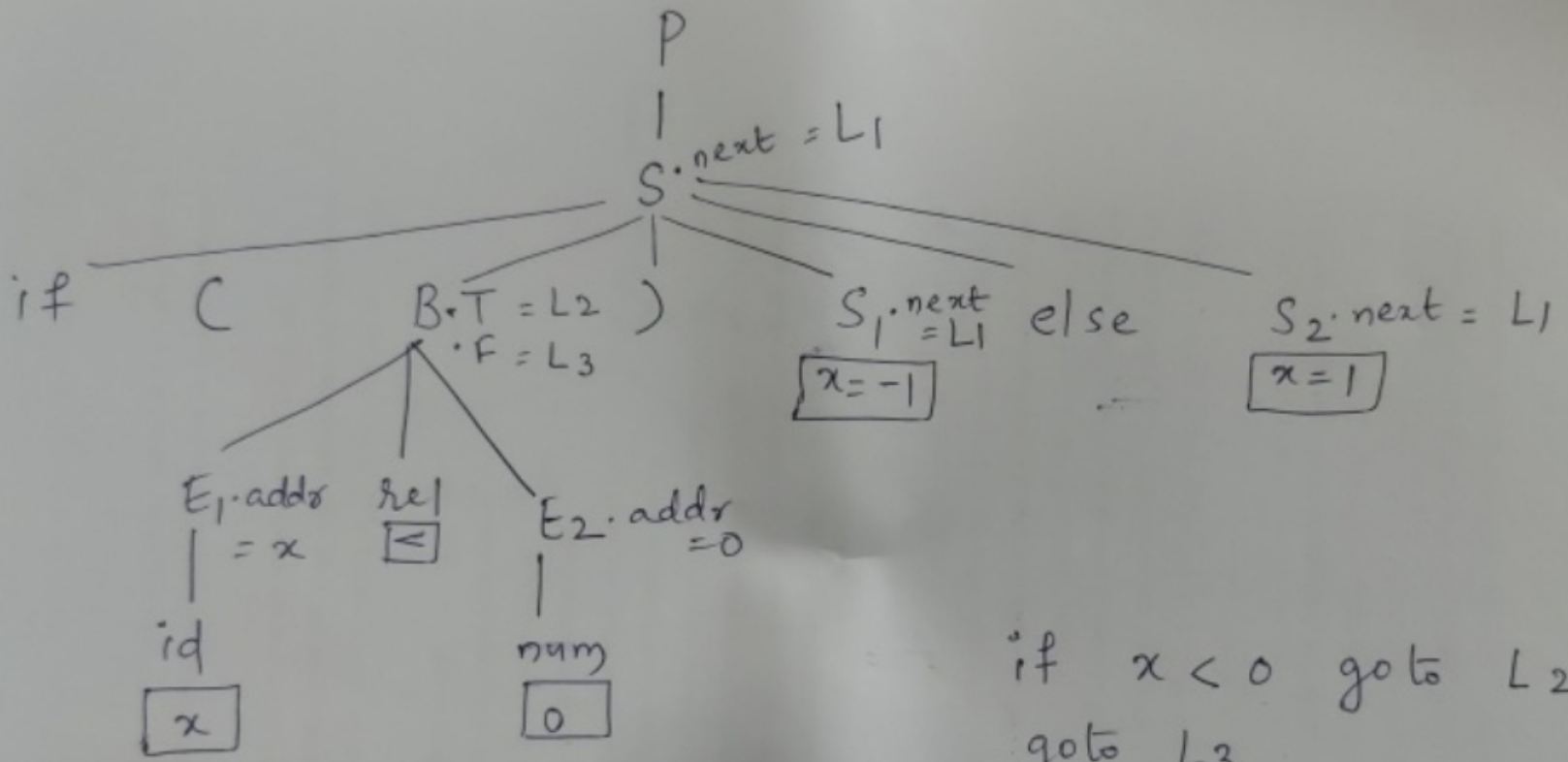
L_2: if x < 0 goto L_3
goto L_1
L_3 : x = -1
goto L_2
L_1 : _ _ _

if (a==b && c==d || e==f) x==1;



if a==b goto L4
goto L3
L4: if c==d goto L2
goto L3
L3: if e==f goto L2
goto L1
L2: x==1
L1: ---

if (x < 0) x = -1 else x = 1 ;



if x < 0 goto L2

goto L3

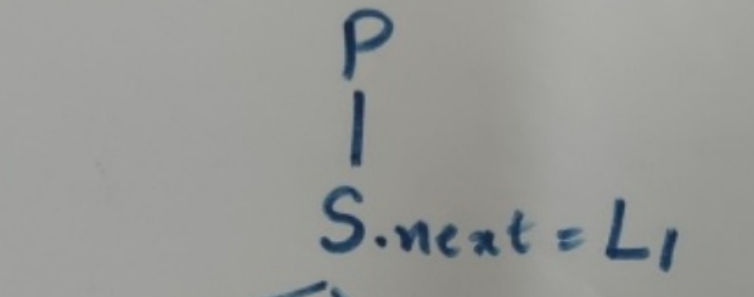
L2 : x = -1

goto L1

L3 : x = 1

L1 : _ _ _

$x = a + b$
$y = x + b$



$S_1.code \Rightarrow$

$x = a + b$

L_2: $S_2.code$
$\Rightarrow y = x + b$

L_1: