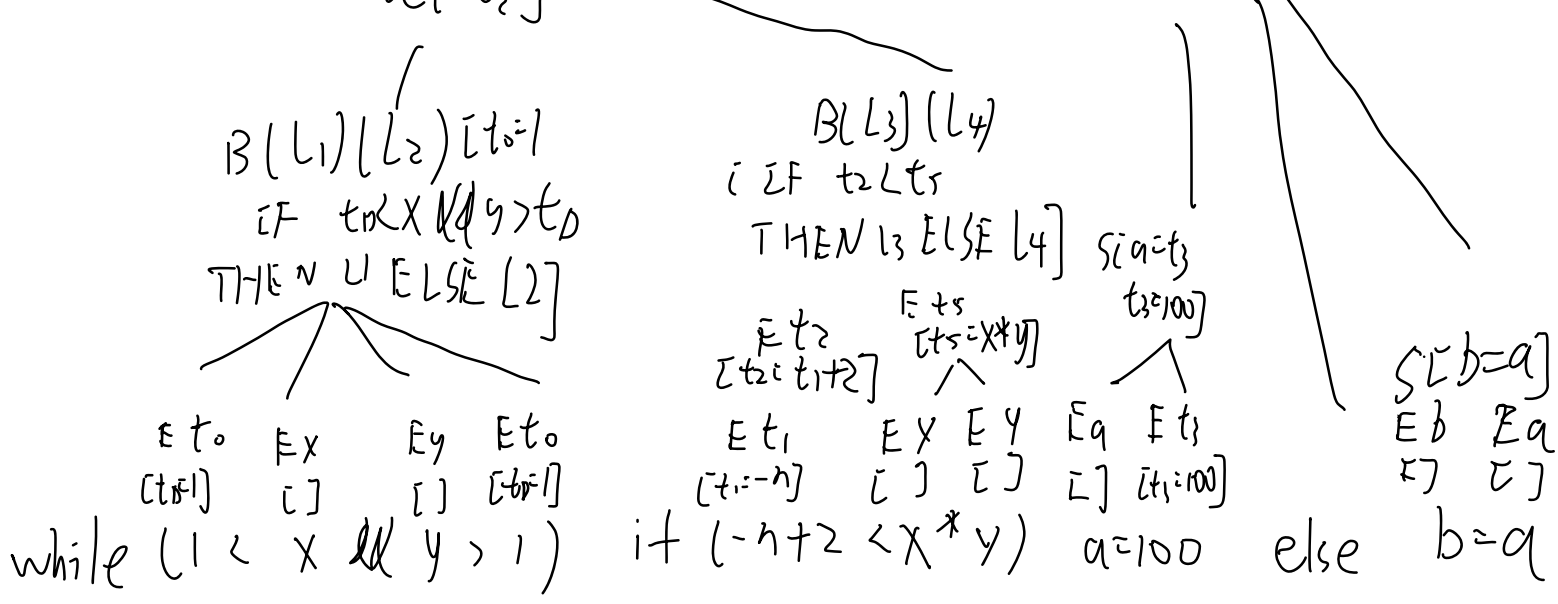
习题 10.2　试翻译下列高级语言语句，写出翻译成的三地址代码段，并给出带注释的语法树。

(1) while(1<x&&y>1)
　　if(−n+2<x*y)a = 100
　　else b = a

(2) if(x) while(y) {
　　if(x>y)goto next;

(1) $t_0 = 1$; IF $t_0 < X$ && $y > t_0$ THEN L1 ELSE L2;
LABEL L1; $t_1 = 0 - n$, $t_2 = t_1 + 2$; $t_3 = X * y$
IF $t_2 < t_3$ THEN L3 ELSE L4;
LABEL L3; $t_3 = 100$; $a = t_3$; GOTO LABEL L5;
LABEL L4; $b = a$; GOTO LABEL L5;
LABEL L5; GOTO L1;
LABEL L2;

S[$t_0 = 1$, if $t_0 < X$ && $y > t_0$, then L_1 else L_2;
label L_1; $t_1 = -n$; $t_2 = t_1 + 2$; $t_3 = X * y$; if $t_2 < t_3$
then L_3, else L_4; label L_3; $a = t_3$; $t_3 = 100$; goto label L5;
label L_4; $b = a$; goto label L5; label L5; goto L_1;
label L_2]



习题 10.3　C 语言的 for 语句的文法 $S \rightarrow \text{for}(S;B;S)S$ 的含义如下：

```
S[1].code
while(B.code) {
    S[3].code
    S[2].code
}
```

试设计属性文法把 C 语言的 for 语句翻译为三地址代码，并把语句

for(i = 0; i<100; i = i+1)print i

翻译为三地址代码段，并给出带注释的语法树作为对翻译过程和结果的解释。

① 属性文法：

S → for (S; B; S) S　{

S[1].code ++

S[3].entry = B.tc; S[1].next = B.fc; S[2].next = S[3].entry

gen[LABEL? B.entry] ++ B.code ++

gen[LABEL? B.tc] ++ S[3].code ++ S[2].code ++

gen[GOTO? B.entry] ++ gen[LABEL? B.fc]

}

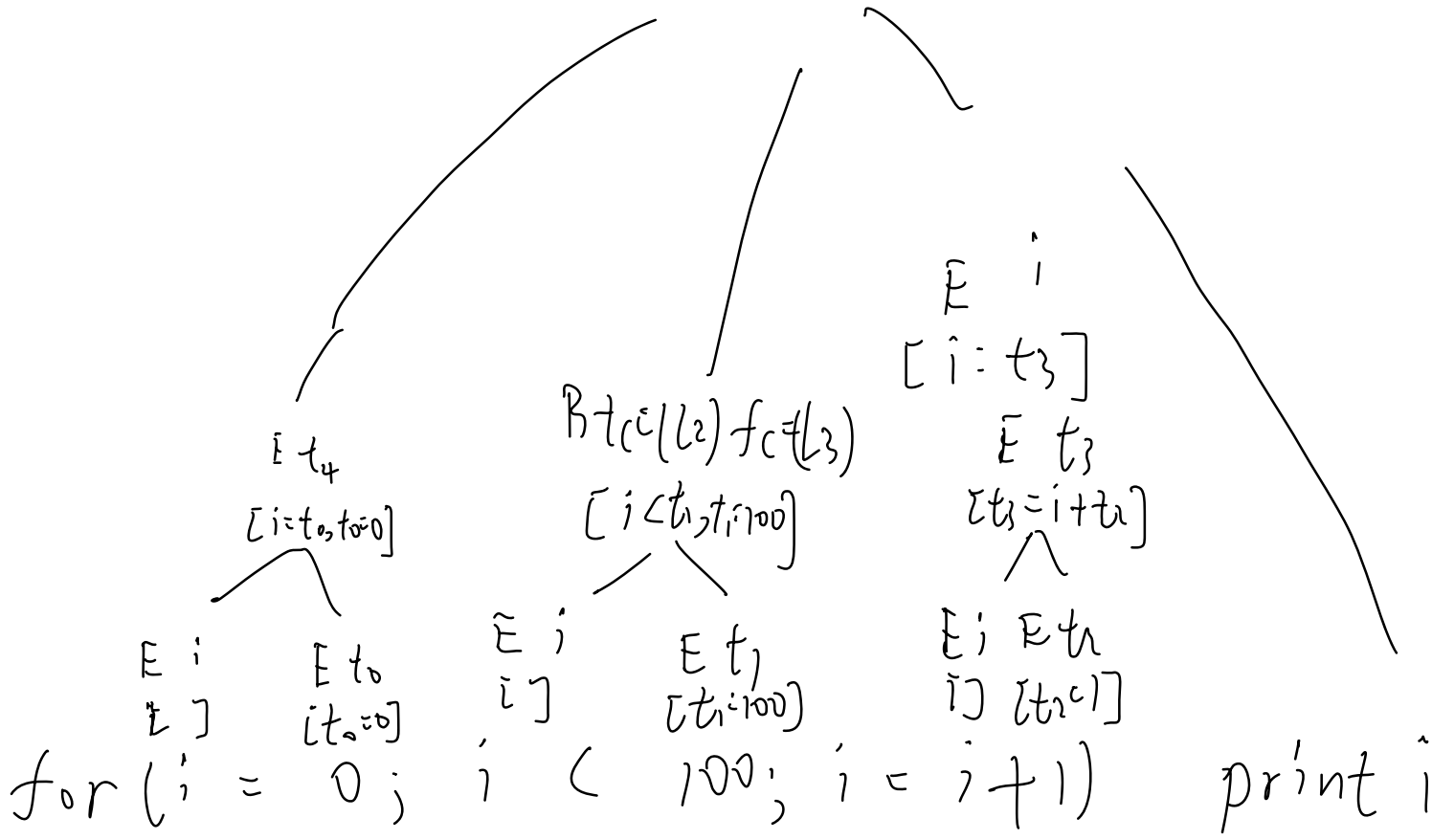
② 三地址代码段：t0 = 0; i = t0

LABEL L1; t1 = 100; if i<t1 GOTO L2

GOTO L3; LABEL L2; PRINT i; t2 = 1; t3 = i+t2

i = t3; GOTO L1, LABEL L3

③ 语法树：S[i=t0, t0=0; label L1; i<t1, t1=100; if i<t1 GOTO L2 GOTO L3; LABEL L2; PRINT i; t2=4; t3=i+t2; i=t3; GOTO L1; LABEL L3]



习题 10.5　程序中的声明如下：

```
int x; float z;
int a[10, 20], b[6];
float bar(int brr[6];) {
    float x;
    x = brr[0]+brr[5];
    return x};
float foo(int x; float boo(); int arr[10, 10];) {
    if(x==0) z = sqrt(boo(arr[0, 0],),)
    else return boo(arr[x, x],)};
```

试参照第 10 章中采用全局名的翻译样例写出以下内容：@table、@code、bar@table、bar@code、foo@table、foo@code。

(1) (a) table : (

outer: NULL width: argc:0 arglist: NIL rtype: VOID

code: [ ]

entry: (name: x type: INT offset: 4)

entry: (name: z type: FLOAT offset: 8)

entry: (name: a type: ARRAY base: 808 etype: INT
dims: 2 dim[0]: 10 dim[2]: 20

entry: (name: b type: ARRAY base: 832 etype: INT
dims: 1 dim[0]: 6

entry: (name: bar type: FUNC offset: 840
mytab: bar (a) table)

entry: (name: foo type: FUNC offset: 848
mytab: foo (a) table)

② bar @table: (
outer: @table width: 12 argc: 1 arglist: (brr) rtype: FLO
entry: (name: brr type: ARRPTT offset: 4 etype: INT
dims: 1 dim[0]: 6)
entry: (name: X type: FLOAT offset: 12))

③ bar @code = [
t1 = brr[0] ; t2 = brr[5]; t3 = t1 + t2; X = t3
RETURN X]

④ foo @table = (
outer: @table width: 12 argc: 3 arglist (x boo arr)
rtype: FLO
entry: (name: X type: INT offset: 4)
entry: (name: boo type: FUNPTT offset: 8 rtype: FLOAT)
entry: (name: arr type: ARRPTT offset: 12 etype: INT
dims: 2 dim[0]: 10 dim[1]: 10))

⑤ foo @ code = [
IF X == 0 THEN L1 ELSE L2;
LABEL L1; t1 = arr[0] ; t2 = CALL boo, 1(t1);
t3 = CALL sqrt, 1(t2); Z = t3; GOTO L3;
LABEL L2; t4 = X * 10 ; t5 = t4 + X; t6 = arr[t5];
t7 = CALL boo 1(t6) ; RETURN t7
LABEL L3]