```
sum = k+w+r;
count = 0;
while (count < length+k){
        sum = sum + array[count];
        if (sum > 100 && count<5){
                sum = sum %100;
        }else{
                sum = sum + sum/100 x5;
        }
        count= count +1;
}
```

$P \to S$

$S \to S_1 S_2$

$S \to \text{assign}$

$S \to \text{while} (B) S_1$

$S \to \text{if} (B) S_1 \text{ else } S_2$

$S \to \text{if} (B) S$

* for complex Boolean expressions we need labels.

if $(\underbrace{x>0}_{} \; \&\& \; \underbrace{\text{fact}(x) < 100}_{})\{$

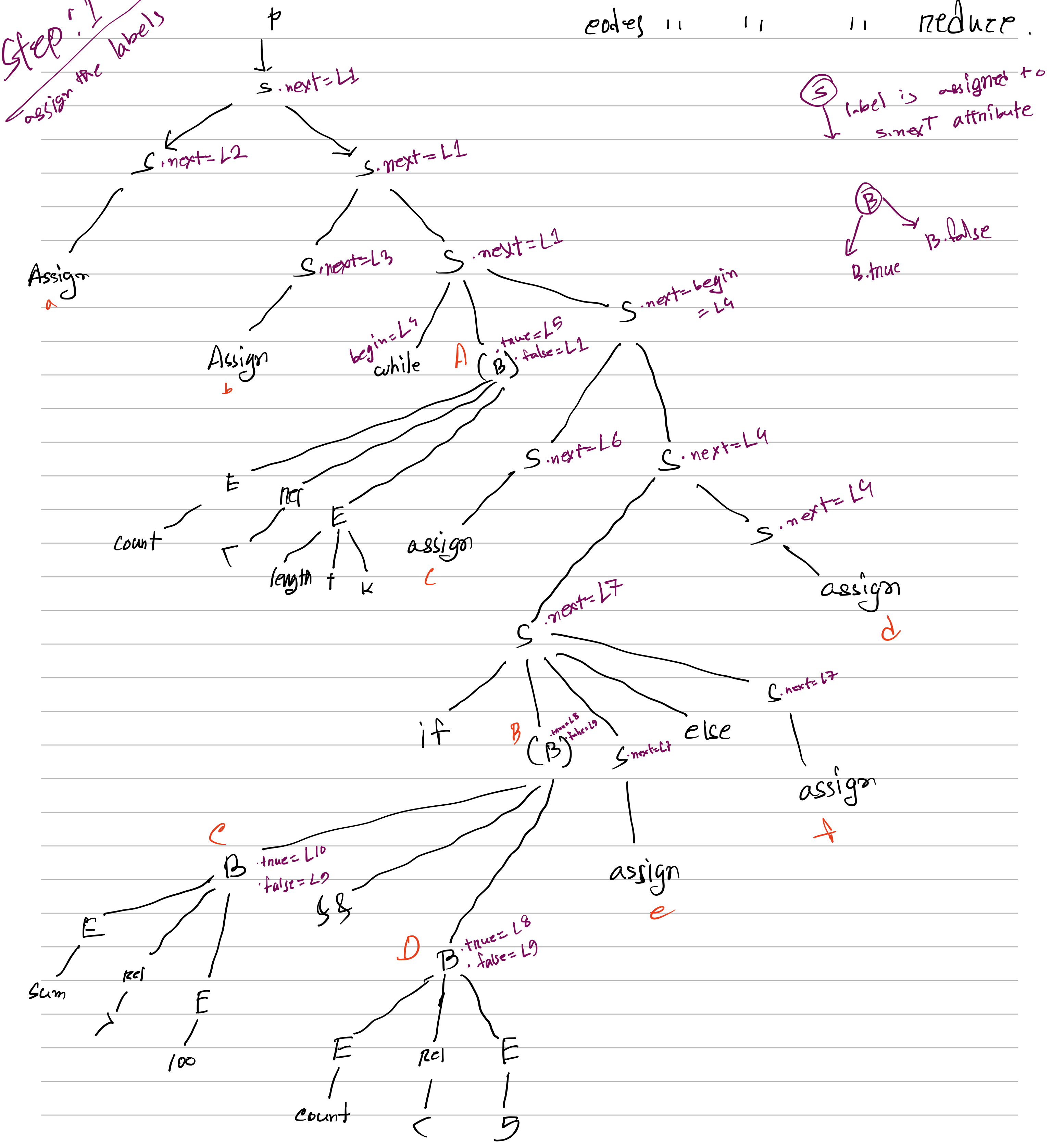
if this is false we dont need to check the part. the function call is required only if the first part is true.

Step : 1

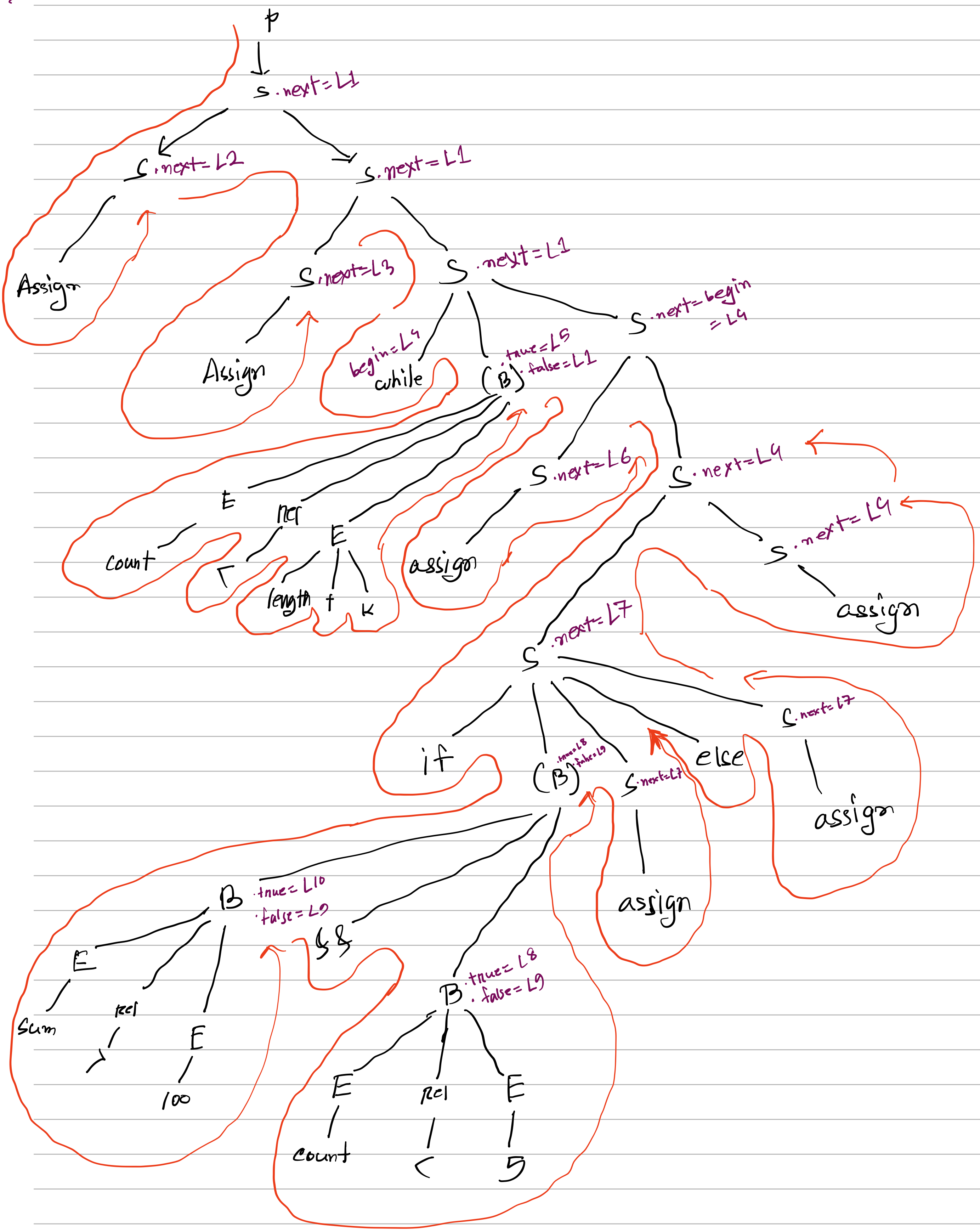
assign the labels

labels are process during shift and

codes ,, ,, ,, reduce.



step 2:


P
S.next=L1
S.next=L2
S.next=L1
Assign
S.next=L3
S.next=L1
Assign
S.next=begin
=L4
begin=L4
while
(B)
.true=L5
.false=L1
E
rel
E
count
Γ
length
f
k
S.next=L6
S.next=L4
assign
S.next=L4
assign
S.next=L7
if
(B)
.true=L8
.false=L9
S.next=L7
else
S.next=L7
assign
assign
B .true=L10
.false=L9
&&
B .true=L8
.false=L9
E
rel
E
sum
>
100
E
Rel
E
count
<
5

a. code

```
t1 = k+w
t2 = t1+n
sum = t2
```

b. code

```
count = 0
```

A. code

```
t3 = length + k
if count < t3 goto L5
goto L1
```

c. code

```
t4 = count * 8    → assuming the array is a float of 10 len.
t5 = array [t4]
t6 = sum + t5
sum = t
```

C. code

```
if sum > 100 goto L10
goto L9
```

D. code

```
if count < 5 goto L8
goto L9
```

B. code

```
if sum>100 goto L10   } B.code
goto L9
L10:        } label (B1.true)
if count<5 goto L8
goto L9
```

```
e.code  | t7 = sum % 100
        | sum = t7
```

```
f.code  | t8 = sum/100
        | t9 = t8 * 5
        | t10 = t9 + sum
        | sum = t10
```

```
d.code  | t11 = count + 1
        | count = t11
```

```
              | if sum>100 goto L10      }
              | goto L9                  }
              | L10:                     }  B.code
              |   if count<5 goto L8     }
              |   goto L9
              |
if else part  | L8:
              |   t7 = sum % 100         }  S1.code
              |   sum = t7
              |
              |   goto L7
              |
              | L9:
              |     t8 = sum/100         }
              |     t9 = t8 * 5          }  S2.code
              |     t10 = t9 + sum       }
              |     sum = t10
```

complete the code