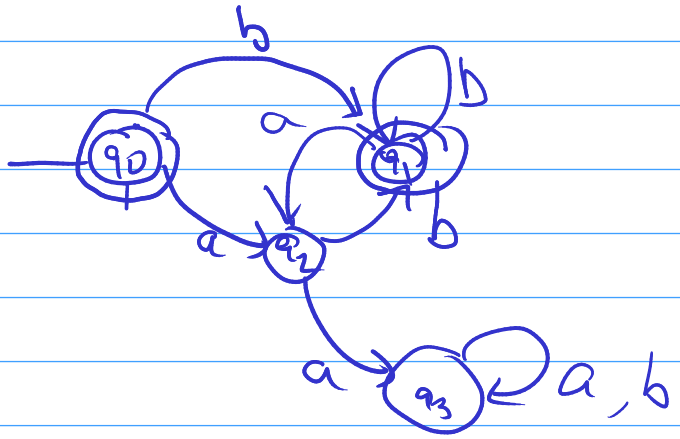
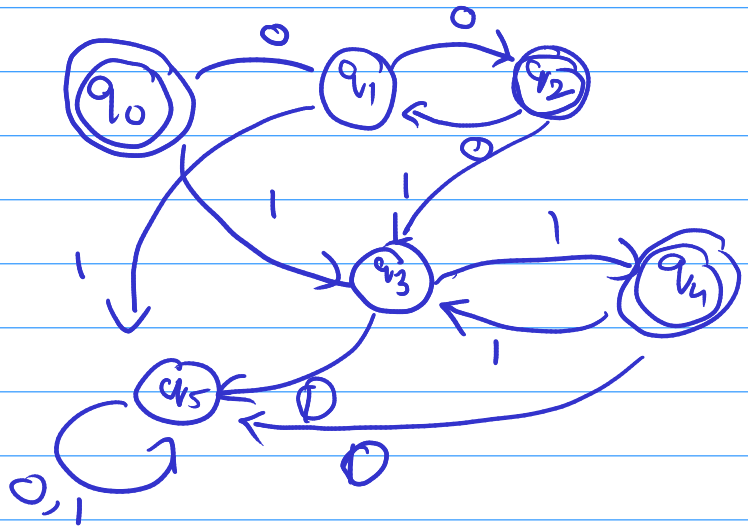
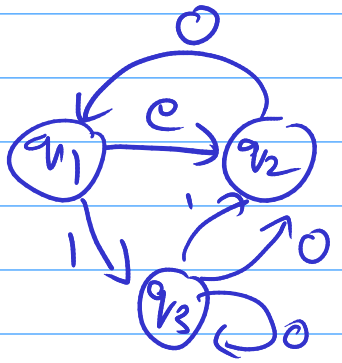




	0	1	ε
q_1		q_3	$q_2\ q_1$
q_2	q_1		q_2
q_3	$q_3 q_2$	q_2	q_3
$q_3 q_2$	q_1		$q_3 q_2$



12	int literal	popl17 -	identifier
+	Operator	— -	Op
a	identifier	1 -	Literal
++	Operator	+ -	Operator
+	Operator	fif -	identifier
==	Operator		
42	literal		

```
a, b, res=0, sigh
if(b==0) lite;
else if((a>0&&b>0) || a<0 && b<0)
    sign=0;
else { sign=1;
    if(a>0) a=-a;
  } elif (b<0) b=-b;
if(a==0) return 0;

while(a>=b){
  a-=b;
  res+=1;
}
int fbcount = 19;
while(fbcount!=0 && a!=0){
  a*=10; while(a>=b){
           a-=b;
           res+=1/10^fbcount);
  fbcount--;
}
```

$S \to L \mid LK$

$K \to A \mid AK$

$A \to \_L \mid \_d \mid KL \mid d$

$L \to a \cdots ZA \cdots Z$

$d \to 0 \cdots 9$

$S = (l)(l \cup d \cup \_l \cup \_d)^*$

$\equiv$

(B) <TERM2> ::= <TERM> ** <TERM> |
<TERM2> ** <TERM> |
<TERM2> ** <TERM2> |
<factor>

<EXP>
A ::= ,A | <EXP> | A, <exp>
max = max(A)

<factor> ::= id | number | (<EXP>) | (max(A A))

max(17, 6-42, 5)

<max> := max(<exp>) |

max(<exp>, <max>[1])

<max>.val = if (max[1] && max[1].val > exp.val)
max[1].val
else
exp.val

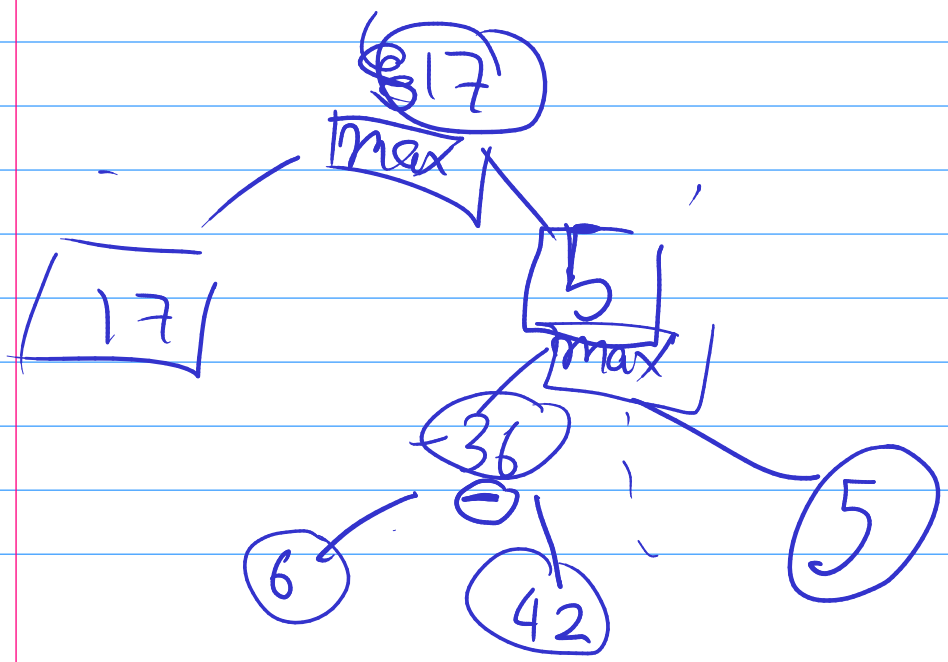
max (17, 6-42, 5)

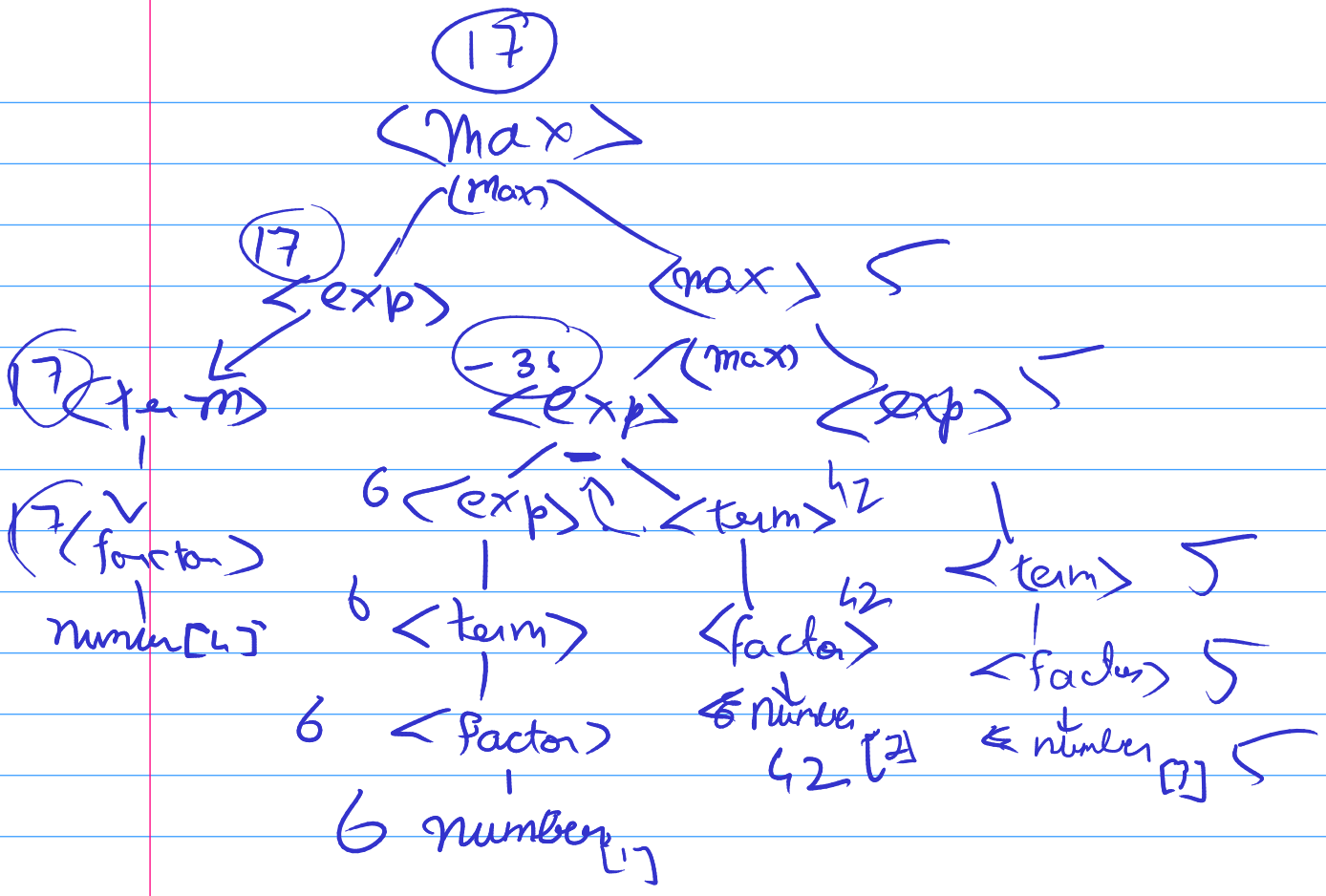
⇒ max(17, max(6-42, 5))

⇒ max(17, max(-36, 5))

⇒ max(17, 5)

⇒ 17





numbers[i].val = Lookup(number[i])

number[1] = 6
number[2] = 42
number[3] = 5
number[4] = 17

expected_type <

or

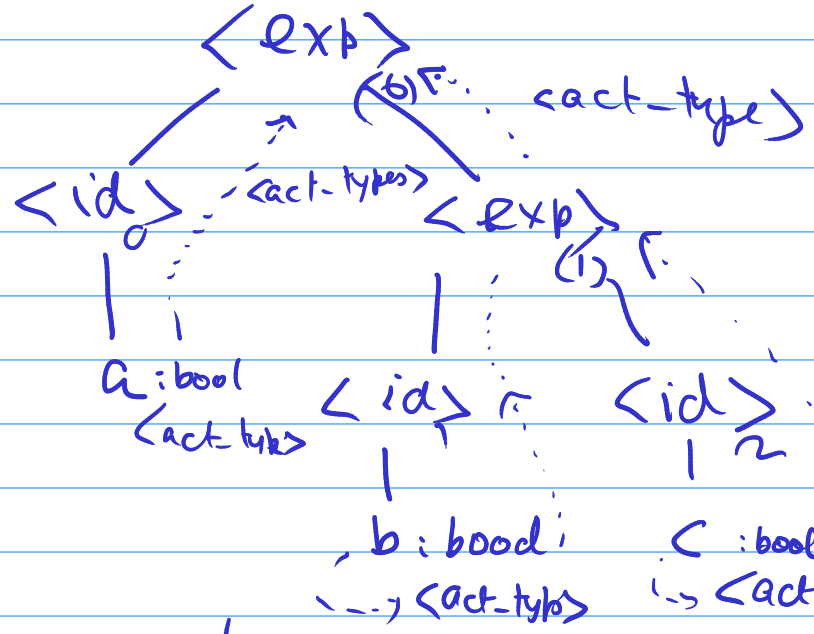
<exp> = id_0 | id_1 OR <exp>[1] | id_2 and <exp>[2]

id = a | b | c

id[i].actual_type = lookup(id)

exp.expected_type = id_0.actual_type OR [i]

<exp>[1].expected_type =



a, b, c ...

<id>[i].act_type = lookup(a) ...

<exp>[1].act_type = if (id(1).act ...

bool

else

error

<exp>[0].act_type = if (id(0) ... (exp)

bool else error.