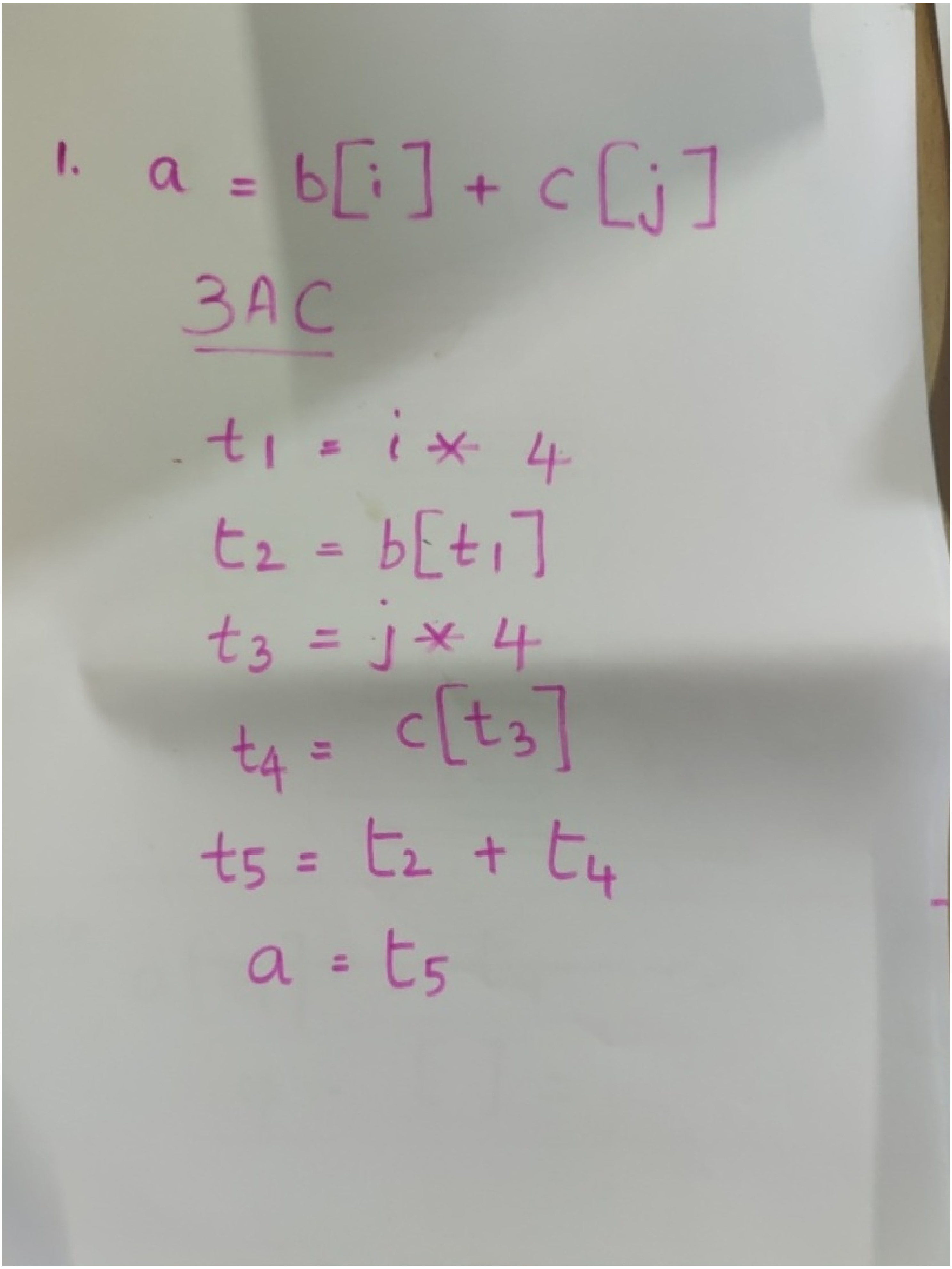


1. $a = b[i] + c[j]$

3AC

$t_1 = i \times 4$

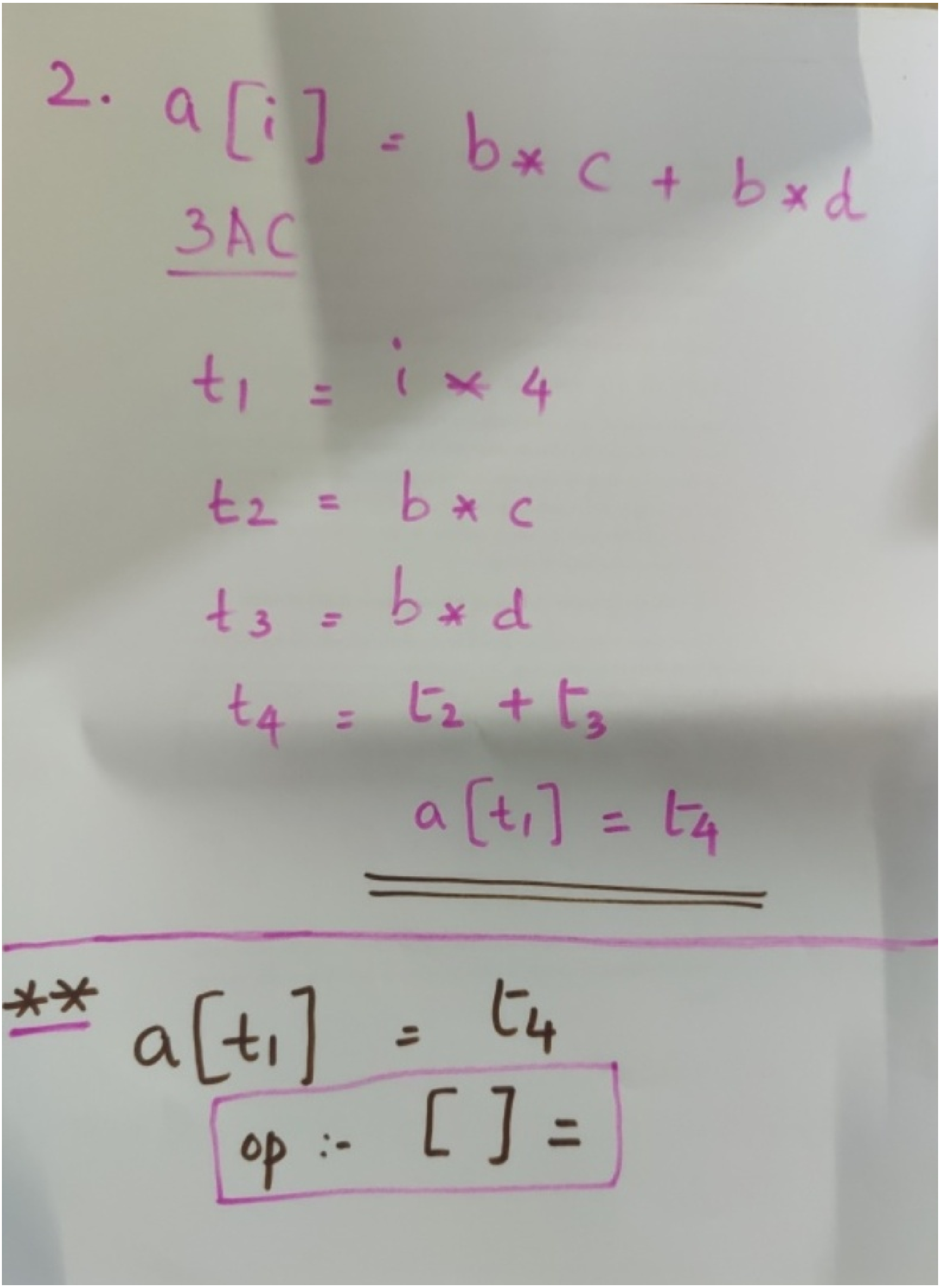
$t_2 = b[t_1]$

$t_3 = j \times 4$

$t_4 = c[t_3]$

$t_5 = t_2 + t_4$

$a = t_5$



2. $a[i] = b * c + b * d$

3AC

$t_1 = i * 4$

$t_2 = b * c$

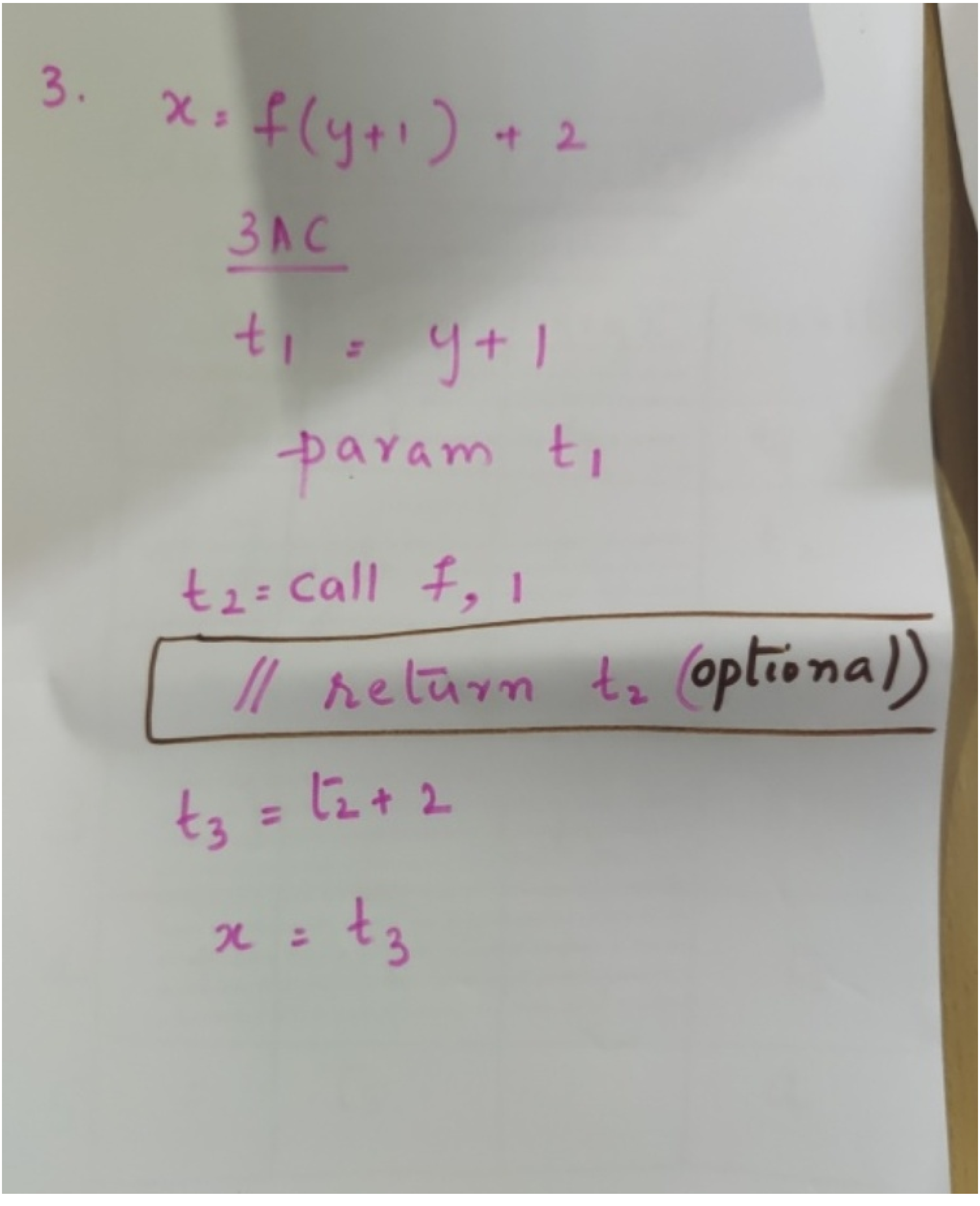
$t_3 = b * d$

$t_4 = t_2 + t_3$

$a[t_1] = t_4$

** $a[t_1] = t_4$

$\boxed{\text{op :- [ ] =}}$



3. $x = f(y+1) + 2$

<u>3AC</u>

$t_1 = y+1$

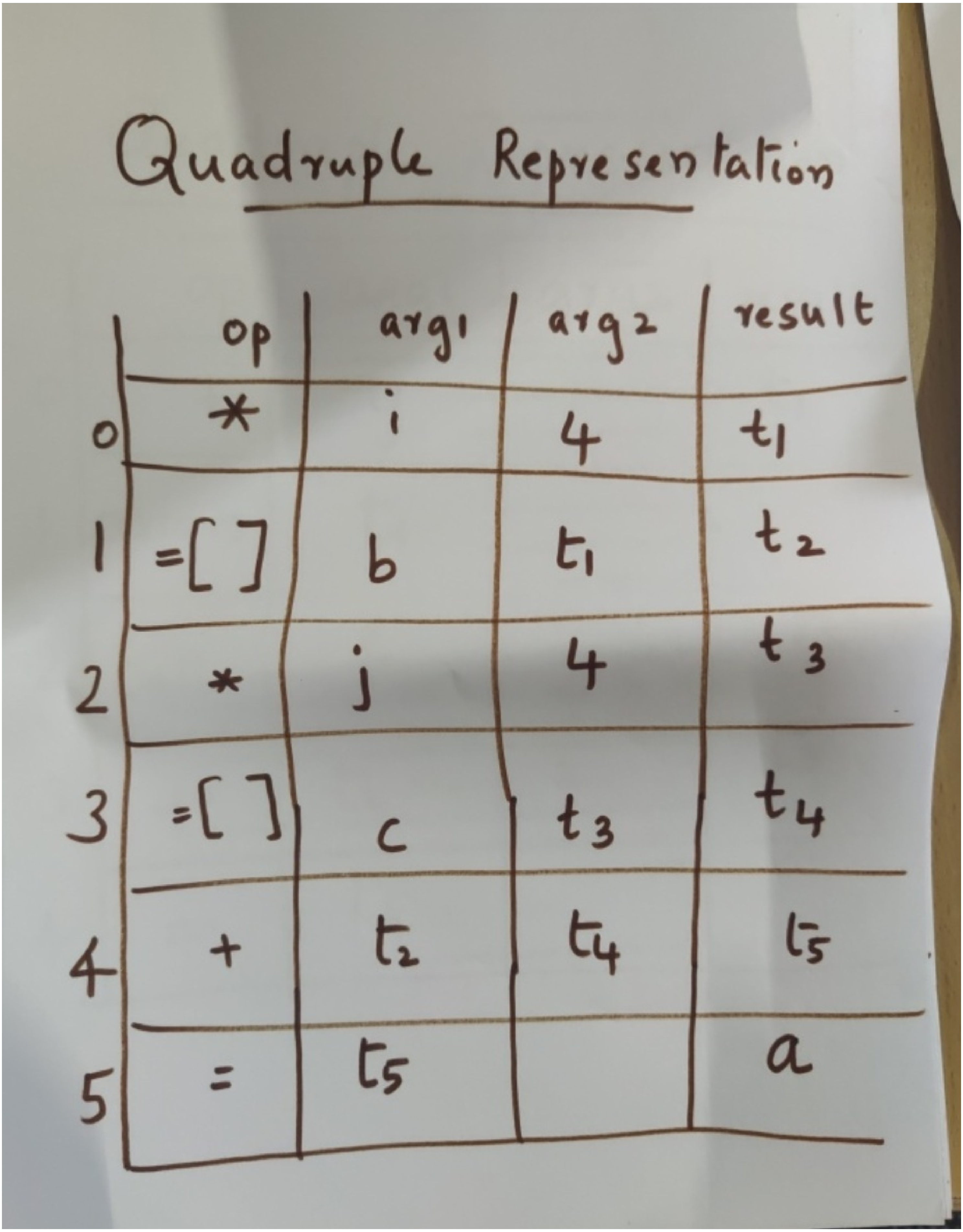
param t_1

$t_2 =$ call $f, 1$

// return t_2 (optional)

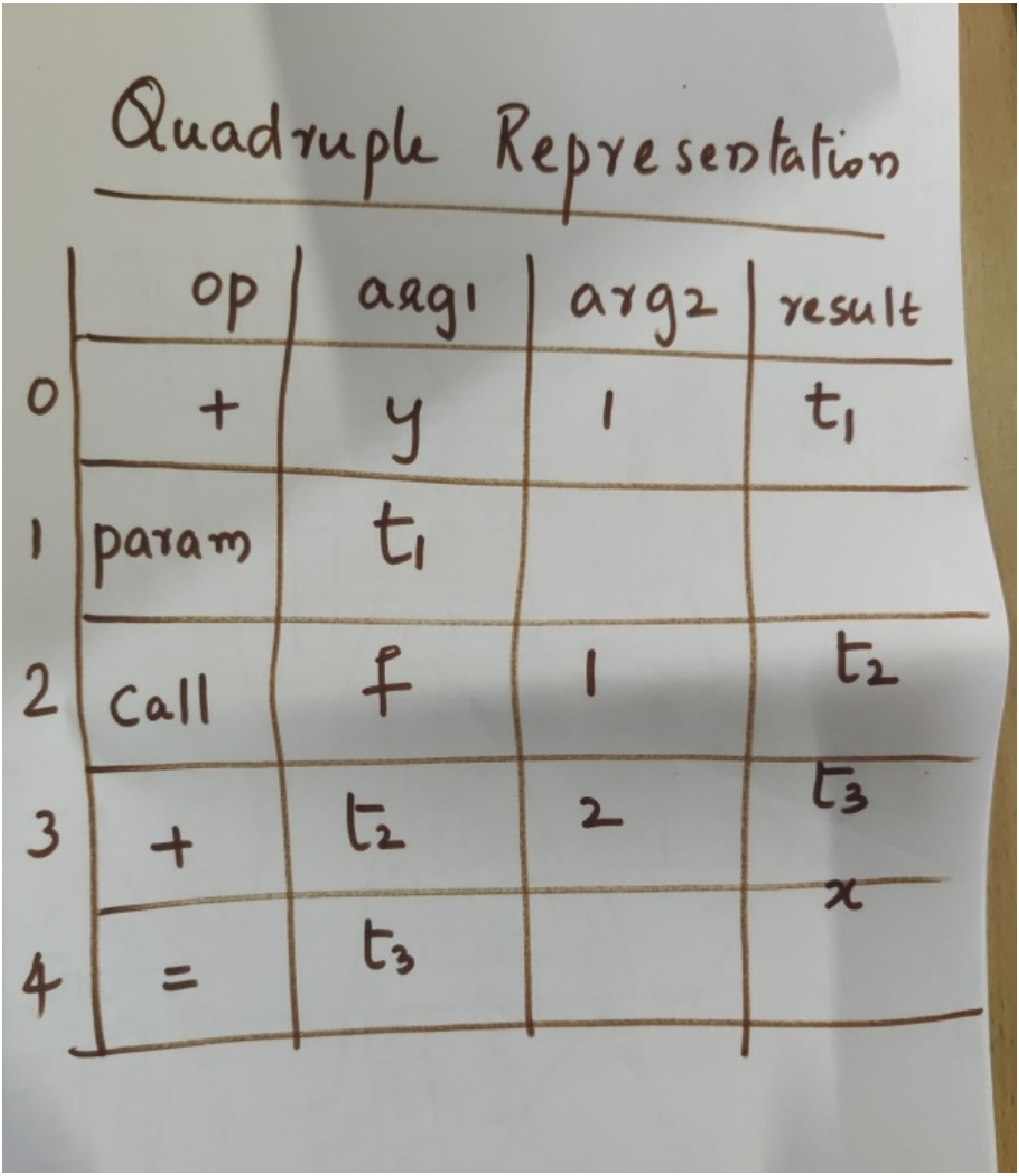
$t_3 = t_2 + 2$

$x = t_3$



Quadruple Representation

	op	arg1	arg2	result
0	*	i	4	t_1
1	=[ ]	b	t_1	t_2
2	*	j	4	t_3
3	=[ ]	c	t_3	t_4
4	+	t_2	t_4	t_5
5	=	t_5		a

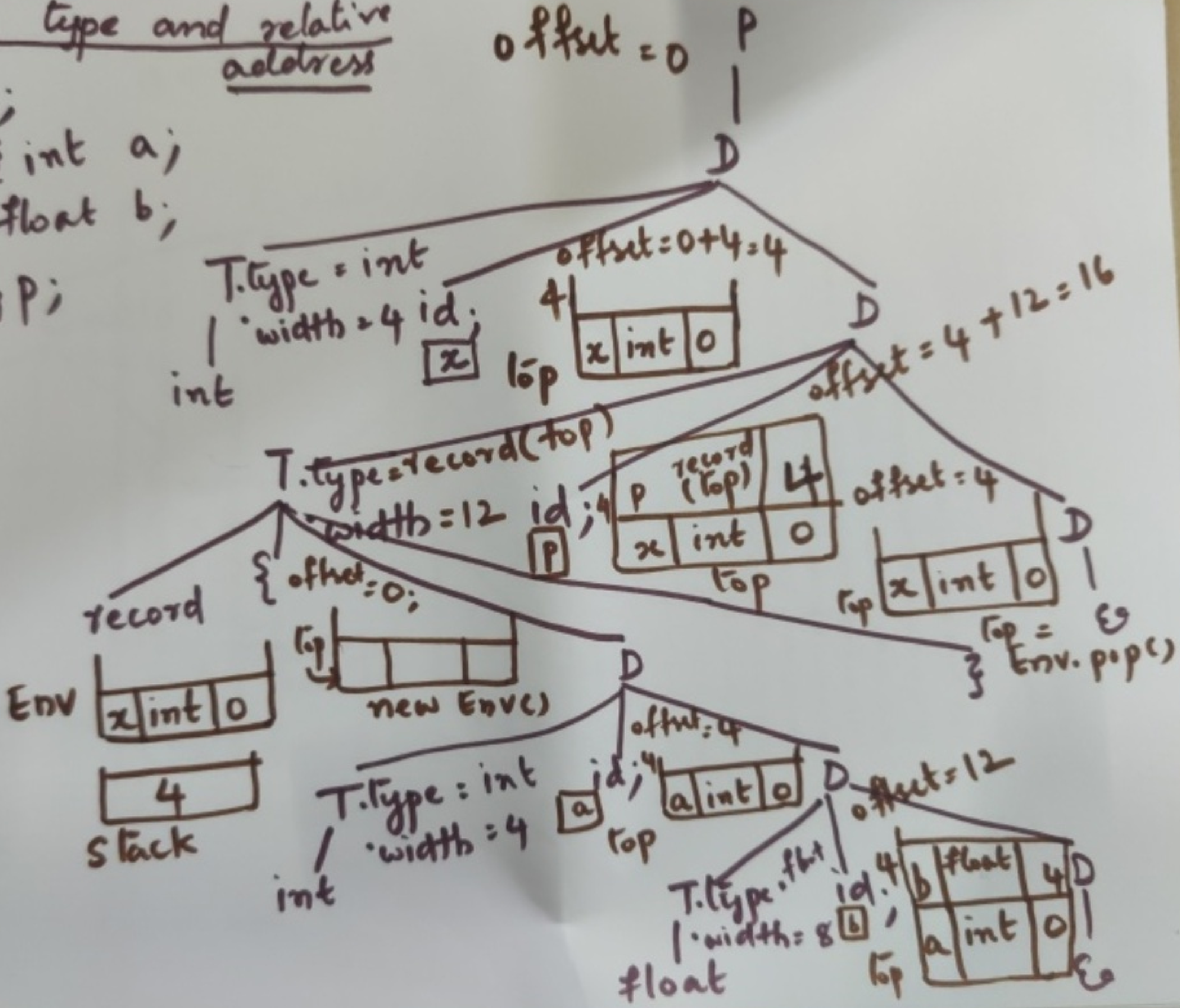


Quadruple Representation

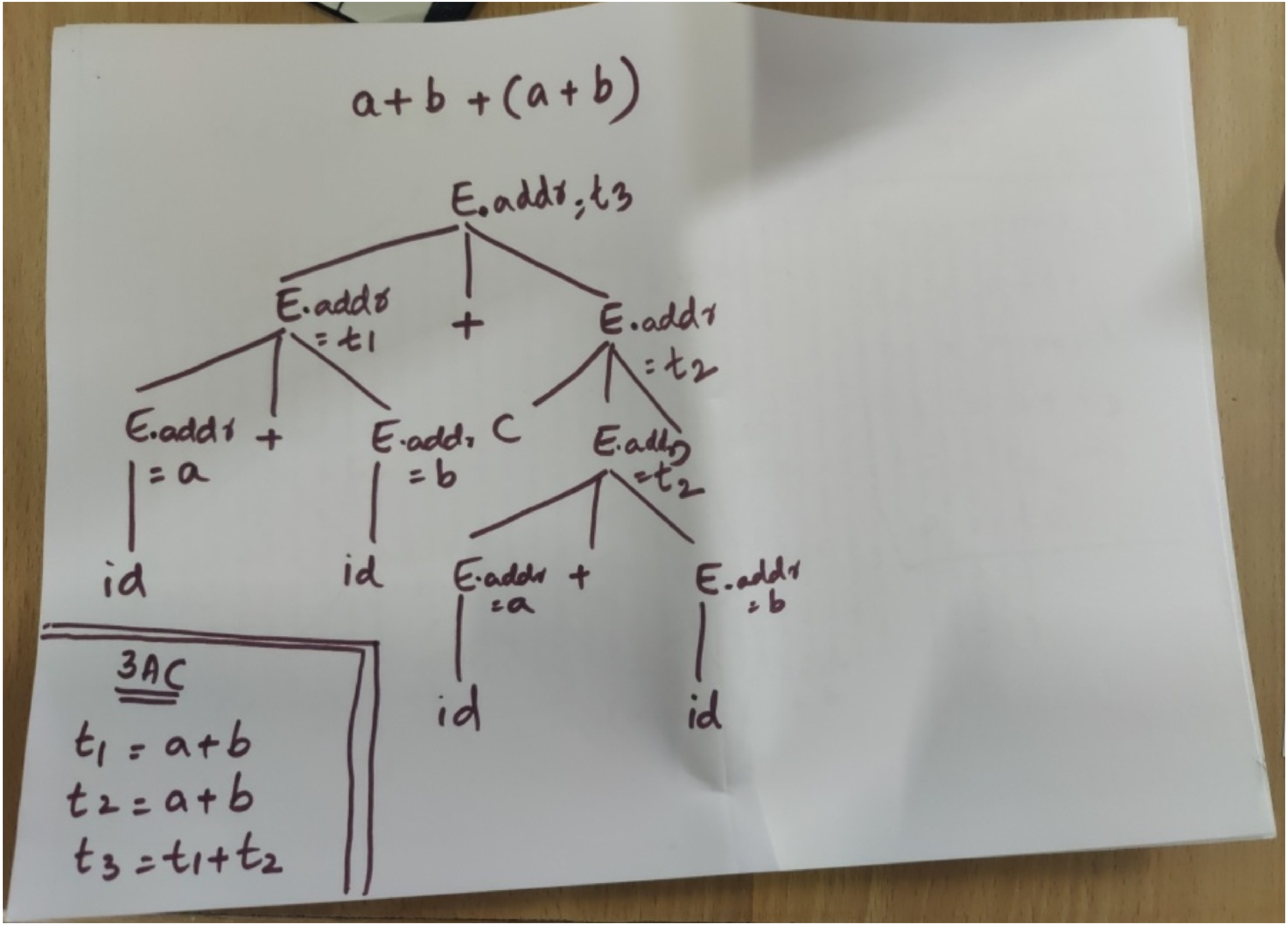
	op	arg1	arg2	result
0	+	y	1	t_1
1	param	t_1		
2	Call	f	1	t_2
3	+	t_2	2	t_3
4	=	t_3		x

Compute type and relative address

```
int x;
record { int a;
         float b;
       } p;
```



$a+b+(a+b)$

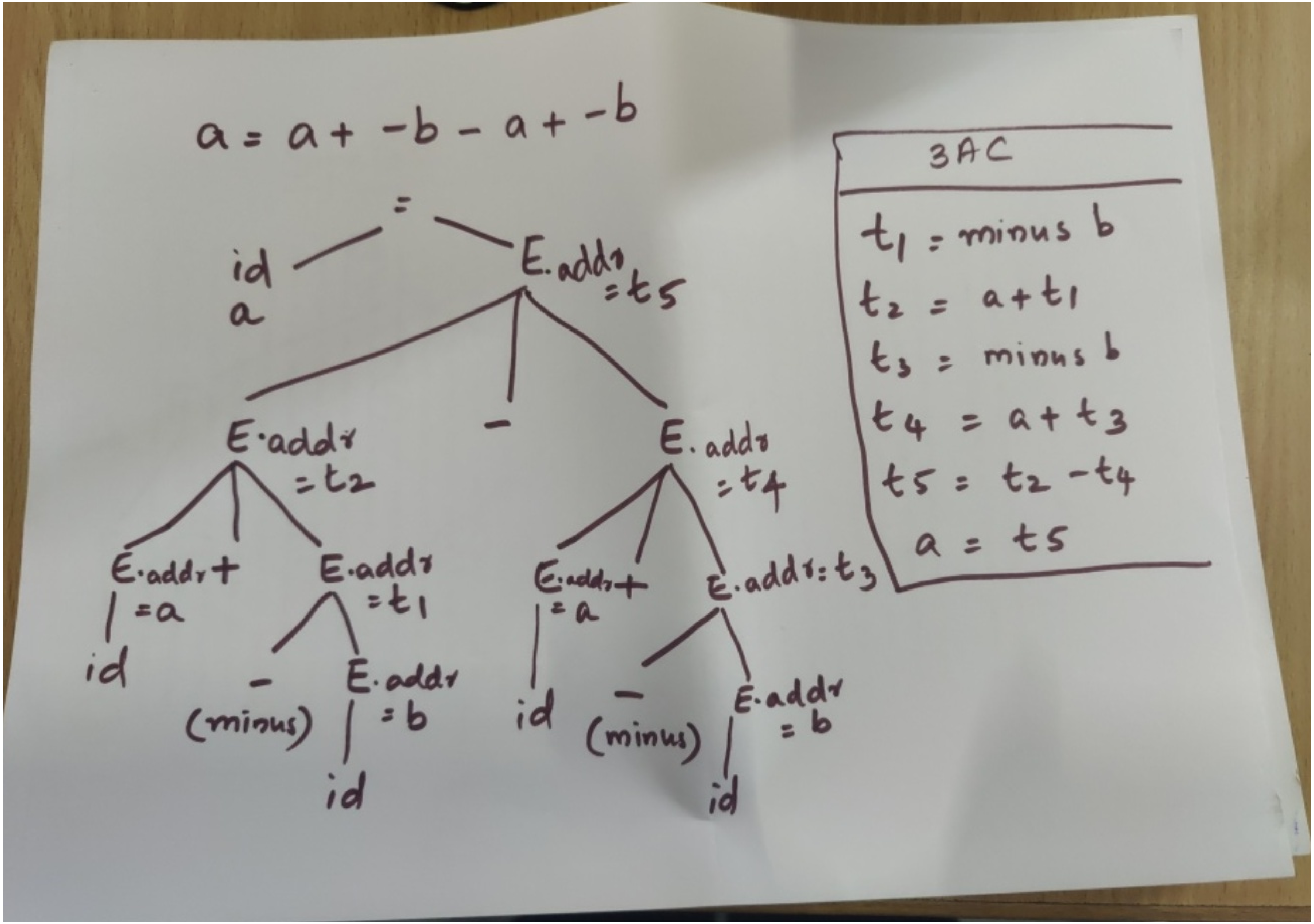


3AC

$t_1 = a+b$

$t_2 = a+b$

$t_3 = t_1+t_2$

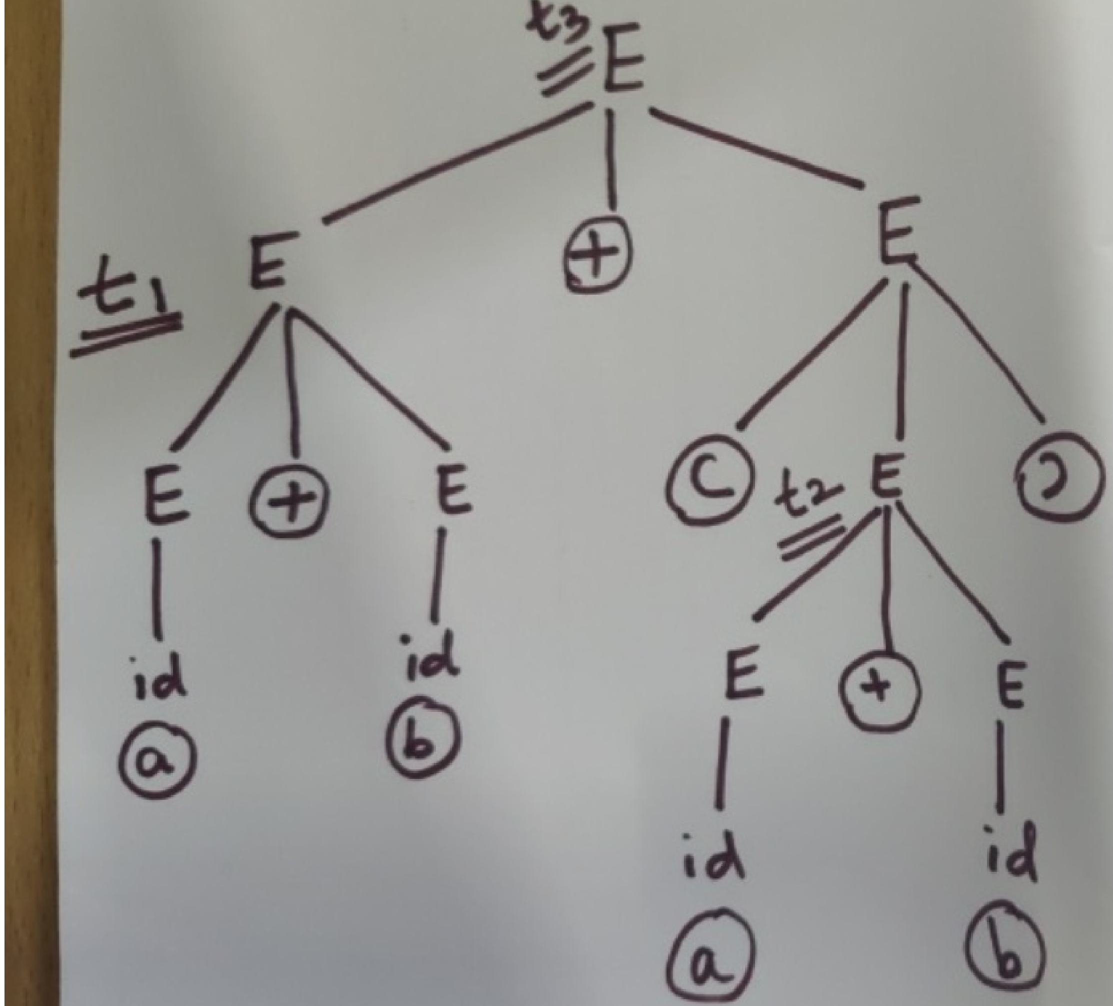


a = a + -b - a + -b

3AC

t_1 = minus b
t_2 = a + t_1
t_3 = minus b
t_4 = a + t_3
t5 = t_2 - t_4
a = t5

Syntax Tree



t3
E
t1
E
+
E
E
+
E
id
a
id
b
(
t2
E
)
E
+
E
id
a
id
b