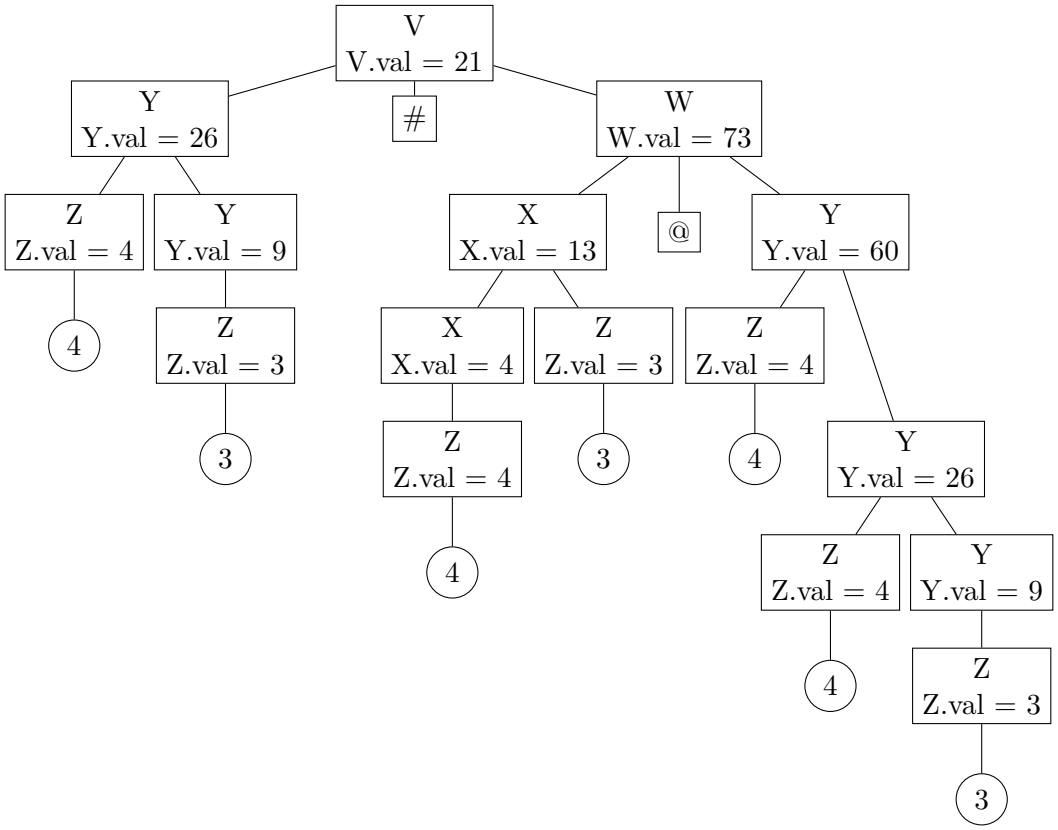
Assignment 3

Sarthak Kalankar
210935

Problem 1

part(i)



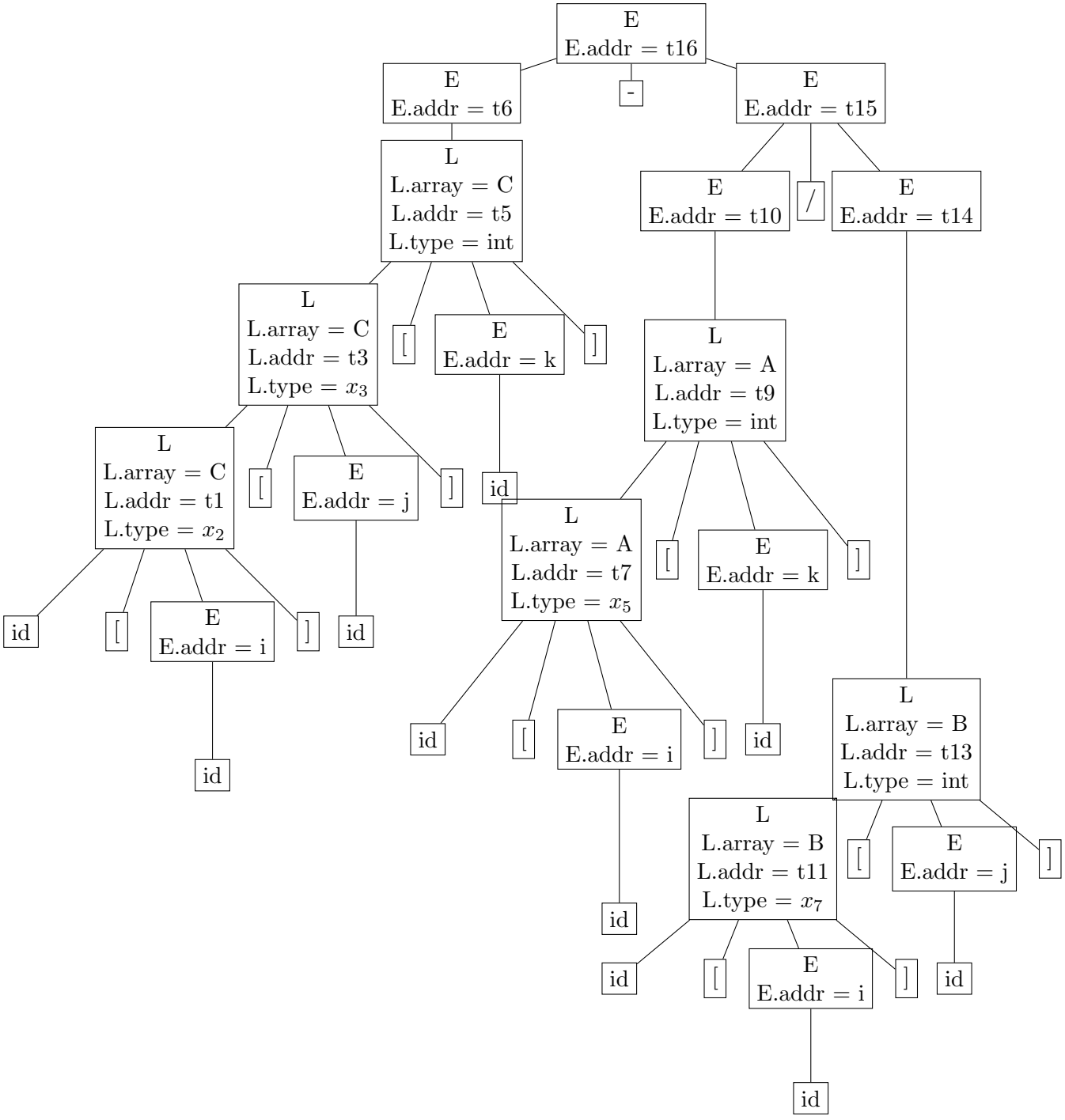
part(ii)

As we can see, the value of V comes out to be 21.

part(iii)

The attribute val of every node has been calculated using the children nodes only. Hence, the grammar is **S-attributed**.

Problem 2



Annotated types

$x_1 = array(10, array(10, array(6, int)$
$x_2 = array(10, array(6, int))$
$x_3 = array(6, int)$

$x_4 = array(11, array(8, int)$
$x_5 = array(8, int)$

$x_6 = array(12, array(6, int))$
$x_7 = array(6, int)$

3AC code

```
t1 = i * 240
t2 = j * 24
t3 = t1 + t2
t4 = k * 4
t5 = t3 + t4
t6 = C[t5]
t7 = i * 32
t8 = k * 4
t9 = t7 + t8
t10 = A[t9]
t11 = i * 24
t12 = j * 4
t13 = t11 + t12
t14 = B[t13]
t15 = t10 / t14
t16 = t6 - t15
```

Problem 3

part (i)

$S \rightarrow \text{id} = E$ {gen(symtop.get(id.lexeme) " = " E.addr)}

$S \rightarrow L = E$ {gen(L.addr = " E.addr)}

$E \rightarrow E_1 + E_2$ {E.addr = newTemp();gen(E.addr " = " E_1.addr " + " E_2.addr)}

$E \rightarrow L$ {E.addr = L.addr}

$L \rightarrow \text{id}$ {L.addr = newTemp(); gen(L.addr = symtop.get(id.lexeme))}

$L \rightarrow \text{id}$ {Elist.array = symtop.get(lexeme.id)}

$[Elist$ {L.addr = newTemp();L.array = symtop.get(id.lexeme) gen(L.addr " = " symtop.get(id.lexeme) "["Elist.addr"]"}

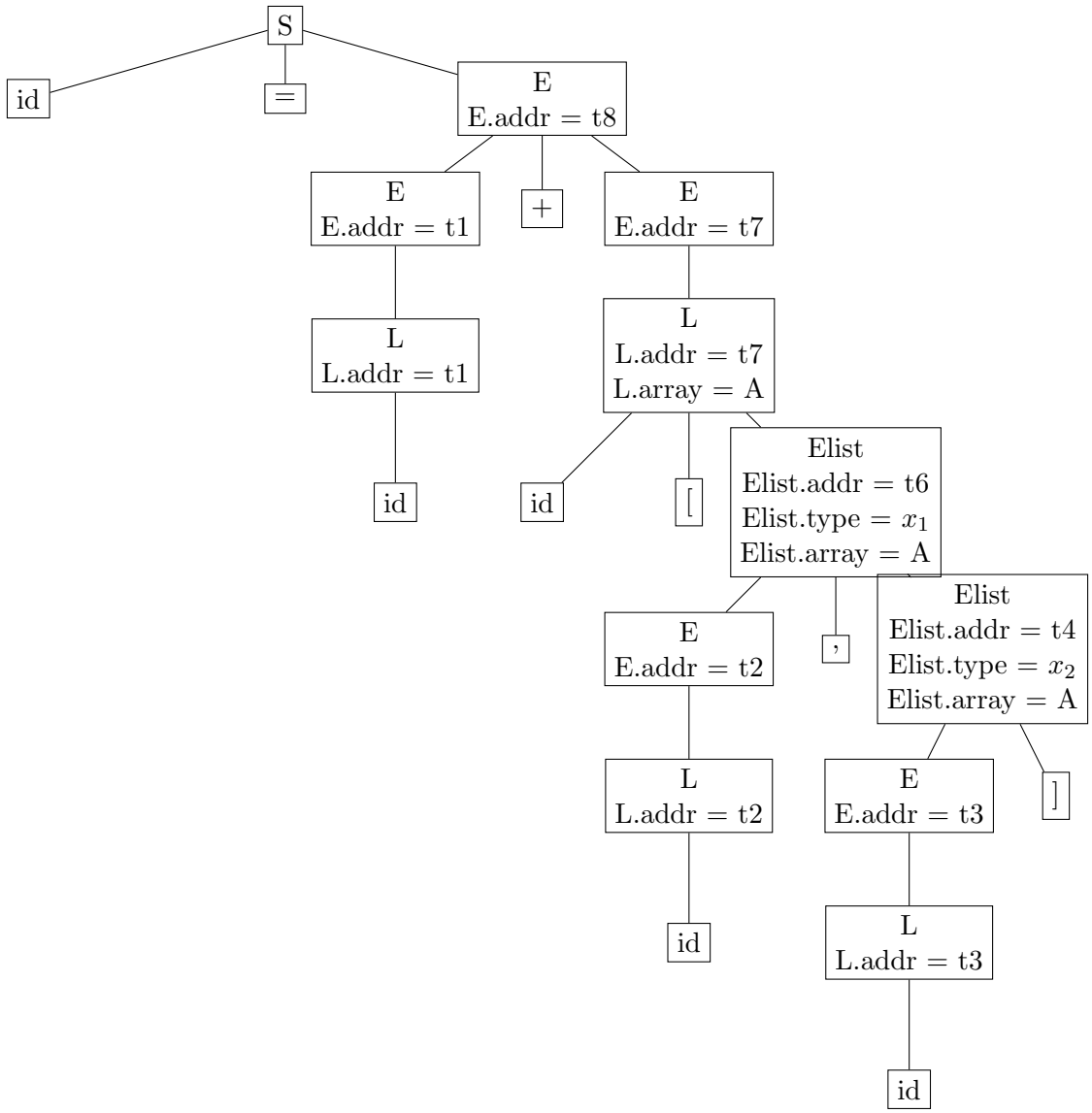
$Elist \rightarrow \text{E}]$ {Elist.addr = newTemp(); Elist.type = Elist.array.type.elem gen(Elist.addr = E.addr " * " Elist.type.width)

$Elist \rightarrow E, Elist_1$ {t = newTemp();Elist.addr = new Temp(); gen(t " = " E.addr " * " Elist1.type.width); gen(Elist.addr " = " $Elist_1$.addr " + " t) Elist.type = $Elist_1$.type.elem)}

part (ii)

- addr keeps track of the temporary assigned to the Non - terminal.
- L.array keeps track of the array name whenever it opens into an array(L goes to id[Elist).
- type keeps track of the type of lexeme or ongoing array.
- width function calculates the size of the type given to it according to the column-major.

part (iii)



Annotated types
$x_1 = array(20, int)$
$x_2 = array(10, array(20, int))$

part (iv)

3AC code

```
t1 = c
t2 = i
t3 = j
t4 = t3 * 40
t5 = t2 * 4
t6 = t4 + t5
t7 = A[t6]
t8 = t1 + t7
x = t8
```