Fokane Sakshi Anil
TE - A - 42

* Consider the input "d=a+b*2" & "sum = a-10/(c*d)" & show output of each phase of compiler.

⟹ There are total six phases of compiler.

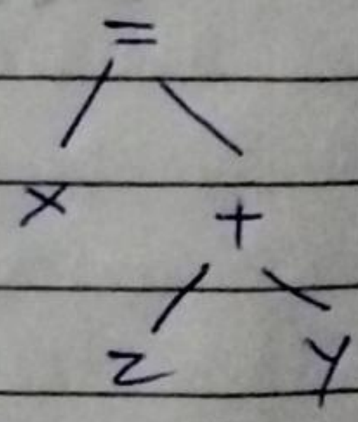
1) Lexical analysis – The lexical analyzer breaks the syntaxes into series of tokens

2) Syntax analysis – It is perform by syntax analyzer. It is also called as parsing. It include recorganization of declaration expression, statements. The purpose of this is to recombine tokens in such a way that is form parse tree / syntax tree.

eg :- $x = z + y$

x, z, y

$=, +$



Parse tree

3) Symantic analysis :- It is used to determine that input has well defined meaning.

4) Intermediate code generation :- The compiler takes input as a source code and generates intermediate code for the machine.

5) Code optimization :- Optimization can be assumed as something that removes unnecessary code lines, and arranges the sequence of statements in order to speed up the program execution without wastage of resources.

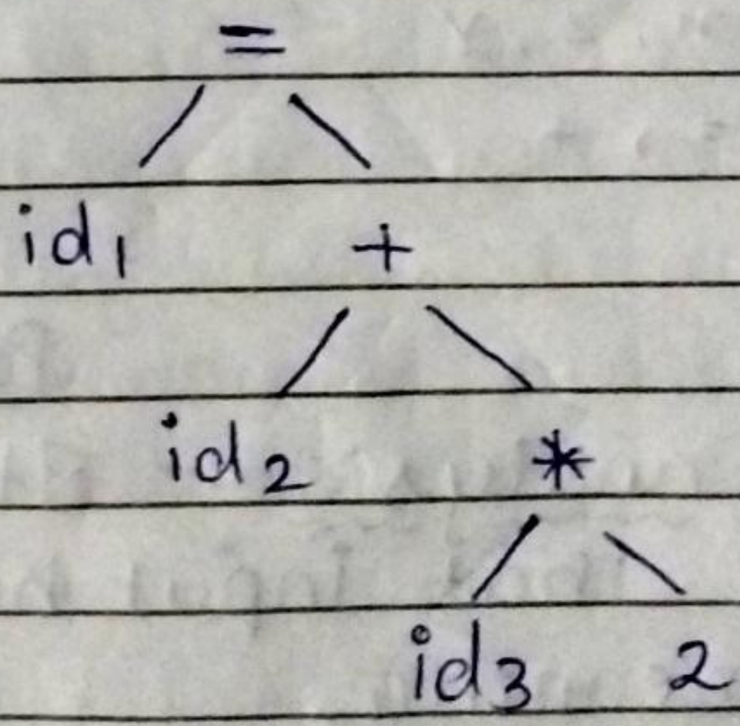
6) Code generation :- The last phase of compiler process is code generation for specific machine. The consideration for this phase memory management, register management, machine specific optimization, machine code.

.... 1) $d = a + b * 2$

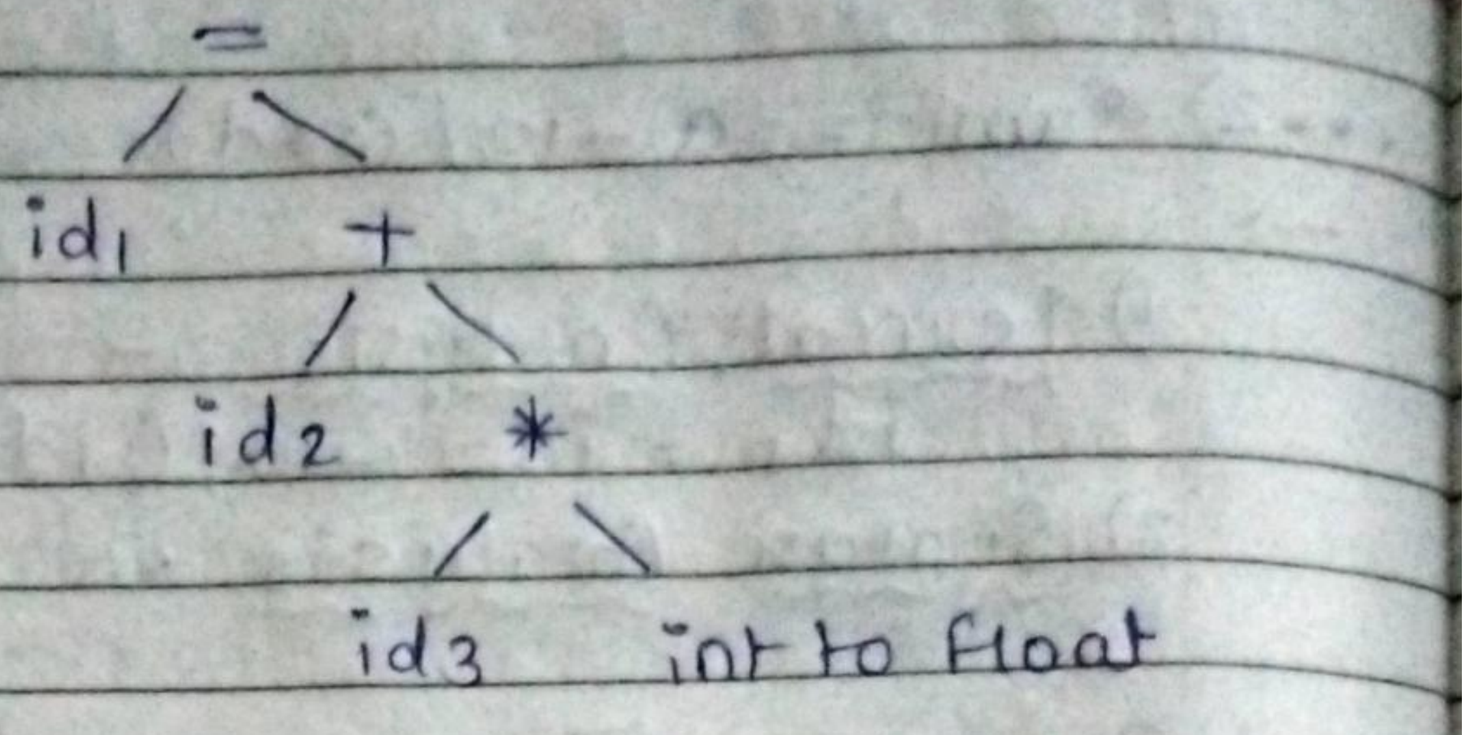
=> 1) Lexical Analyzer :-

$id_1 = id_2 + id_3 * 2$

2) Syntax analyzer :-



3) Semantic analysis :-



4) Intermediate code generation :-

t_1 = int to float(2)
t_2 = id_3 * t_1
t_3 = id_2 + t_2
id_1 = t_3

5) Code optimization :-

t_1 = id_3 * 2.0
id_1 = id_2 + t_1

6) Code generation :-

```
MOV   R1 , id3
MULF  R1 , 2.0
MOVF  R2 , id2
ADDF  R1 , R2
MOVF  id1 , R1
```

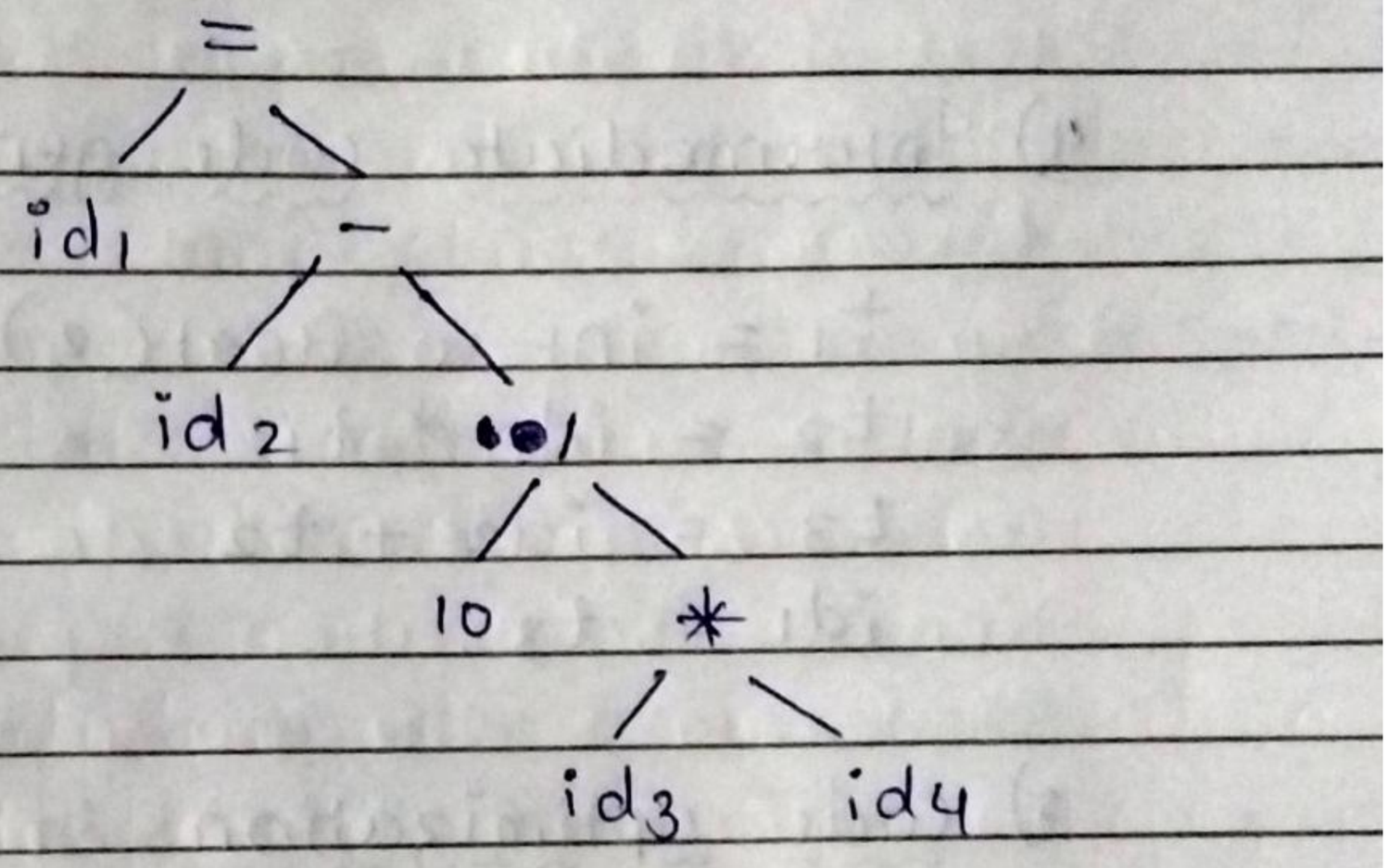
...2) Sum = a - 10 / (c * d)

→

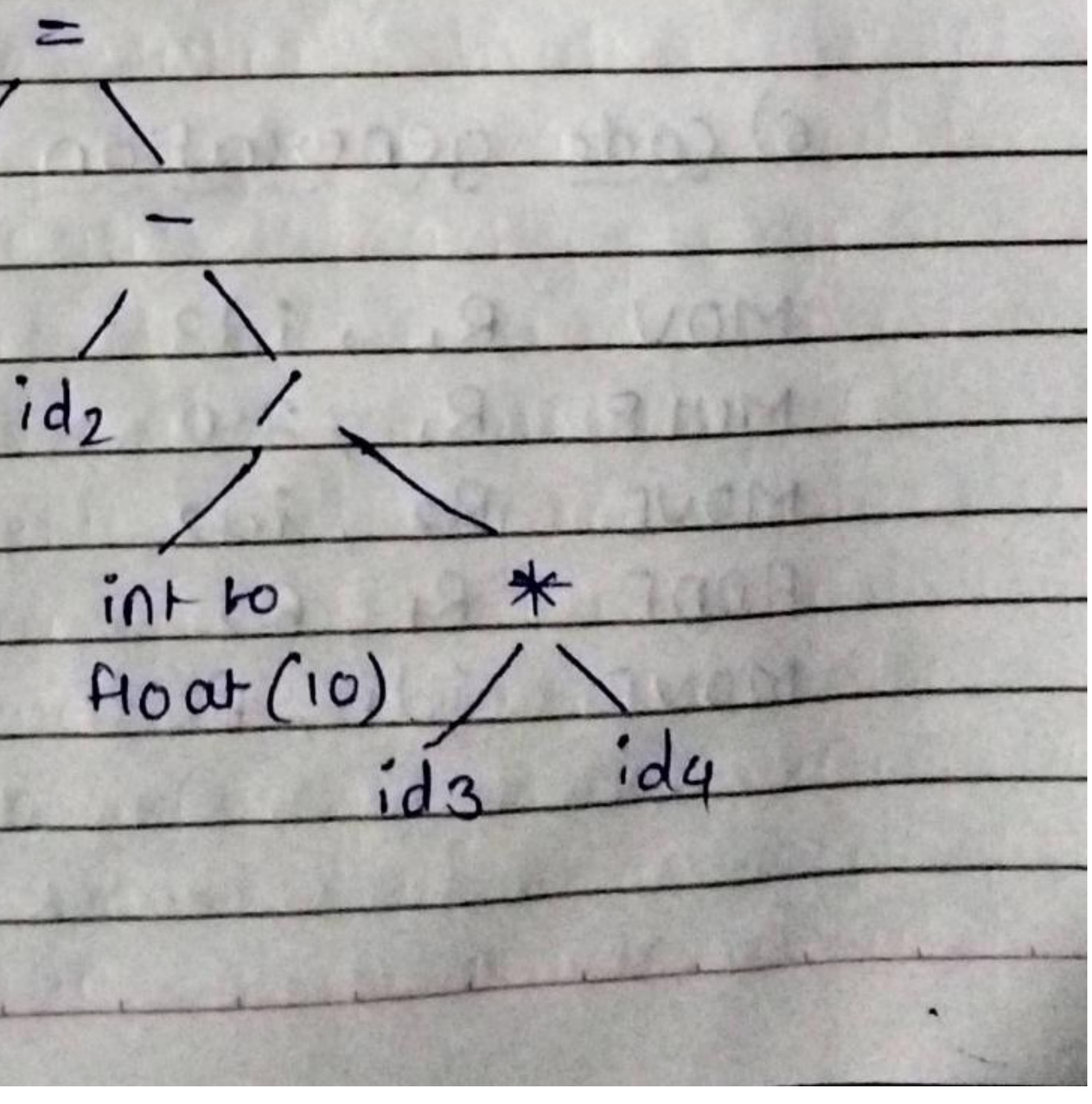
1) Lexical Analysis :-

$id_1 = id_2 - 10 / (id_3 * id_4)$

2) Syntax Analysis :-



3) Semantic analysis :-



4) Intermediate code generation :-

t_1 = int to float(10)
t_2 = id3 ; t_3 = id4
t_4 = id3 - id4
t_5 = t_1 / t_4
t_6 = id2 - t_5
id1 = t_6

5) Code optimization :-

t_1 = int to float(10)
t_2 = id3 * id4
t_3 = t_1 / t_2
t_4 = id2 - id3
id1 = t_4

6) code generation :-

```
MOVF   id3, R2
MULF   id4, R2
DIVF   10.00, R2
SUBF   id2, R3
MOVF   R2, id1
```

- Conclusion :-
Thus we have learned the concept of phasis of compiler.