③ The Phases of Compiler are :

i) Lexical Analysis :- It is the first phase of compiler. It takes source code as input. It reads the source program one character at a time and converts it into meaningful lexemes. It represents lexems in the form of token

ii) Syntax Analysis :- It takes tokens as input and generates a parse tree as output. It also checks the expression made by the tokens is correct or not.

iii) Sementic Analysis :- It checks whether the parse tree follows the rule of language.

iv) Intermediate Code Generation :- In this phase the intermediate code is generated. This code is generated b/w the high level language and low level or machine language.

v) Code Optimization :- This phase helps to improve intermediate code so that the output of program could run faster and take less space. It uses three address code for this ~~for~~ Optimizing the code.

vi) Code Generation :- It is the final phase of compiler. It takes the optimized code as input and produce a output in machine code.

Example → a = b * c - d

Scanned with CamScanner

CAMPUS 2 Date Page

Source Code

↓

I Lexical Analyzer

↓

~~identifier~~

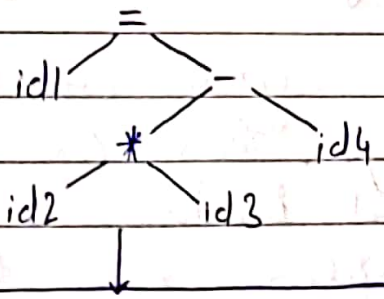
a = b * c - d

id1 — id2 — operator — id3 — id4

↓

II Syntax Analyzer

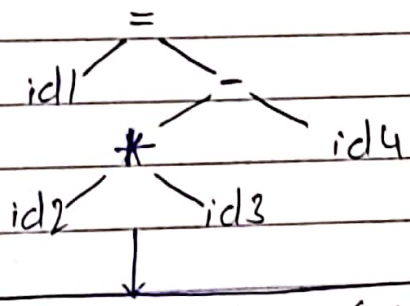
↓



↓

III Semantic Analyzer

↓



↓

IV Intermediate Code Generator

↓

temp1 = id2 * id3

temp2 = temp1 - id4

id1 = temp2

↓

Code Optimization

↓

Scanned with CamScanner

CAMPUS 2 Date Page

temp1 = id2 * id3
id = temp1 - id4

↓

Code Generation

```
MOV   id3, R2
MUL   id2, R2
MOV   id4, R1
SUB   R2, R1
MOV   R1, id1
```

Scanned with CamScanner