1) Compiler

-> It is a translator that converts the high-level language into the machine language.

-> Compiler is used to show errors to the programmer.

-> High-level - developer
machine language - processor

Phases of compiler

1) Lexical analyzer
2) Syntax analyzer
3) Semantic analyzer
4) Intermediate code generation
5) code optimization
6) code generation

1) Lexical analysis

-> first phase of compilation process.
-> source code as input.
-> reads the source program one character at a time and converts it into meaningful lexemes.
-> lexical analyzer represents these lexemes in the form of tokens.
-> < token-name, attribute-value >

Scanned with CamScanner

2) Syntax Analysis

-> second phase of compilation process.

-> takes tokens as input and generates a parse tree as output.

-> The parser checks that the expression made by the tokens is syntactically correct or not.

3) Semantic Analysis

-> Third phase of compilation process.

-> It checks whether the parse tree follows the rules of language.

-> Semantic analyzer keeps track of identifiers, their types and expressions.

-> The output of semantic analysis phase is the annoted tree syntax.

4) Intermediate Code Generation

-> compiler generates the source code into the intermediate code.

-> generated b/w the high-level language & the machine language.

-> you can easily translate it into the target machine code.

Scanned with CamScanner

5) Code Optimization

-> optional phase.

-> impoove intermediate code so that the output of the program could run fast & take less space.

-> removes the unnecessary lines of the code

-> arranges the sequence of statements in order to speed up the program execution.

6) CPU Generation

-> Final stage.

-> takes the optimized intermediate code as input and maps it to the target machine language.

-> translates the intermediate code into the machine code of the specifier computer

Scanned with CamScanner

2) x = a + b * c - d

$\downarrow$

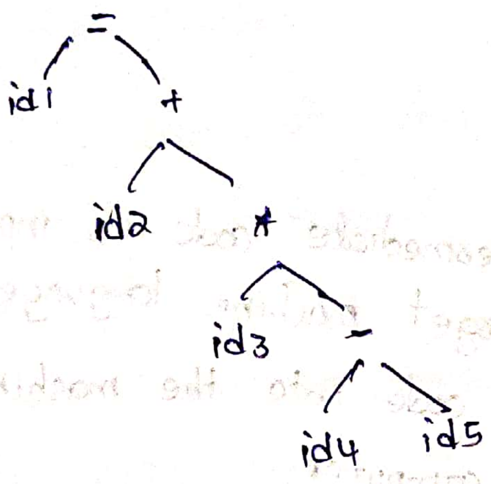
lexical analyser

id1 = id2 + id3 * id4 - id5

$\downarrow$

syntax analysis

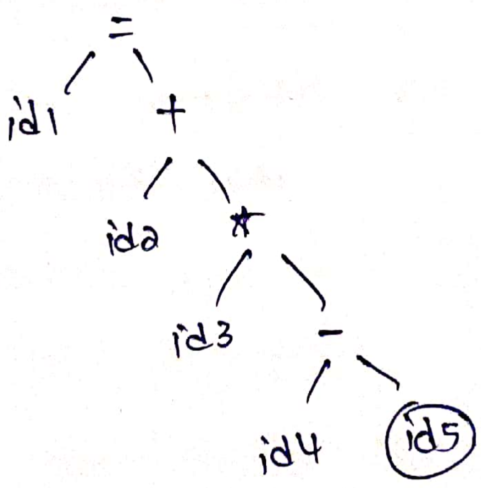
$\downarrow$



$\downarrow$

Semantic Analysis

$\downarrow$



Scanned with CamScanner

$\boxed{\text{Intermediate code generator}}$

↓

t1 = id5

t2 = id4 - t1

t3 = id3 * t2

t4 = id2 + t3

id1 = t4

↓

$\boxed{\text{Code optimization}}$

↓

t2 = id4 - id5

id1 = id2 + t3

t3 = id3 * t2

↓

$\boxed{\text{Code Generator}}$

↓

```
Mov   R4, id5
Mov   R3, id4
Sub   R3, R4
Mov   R2, id3
Mov   R2, R3
Mov   R1, id2
Add   R1, R2
Mov   id1, R1
```

Scanned with CamScanner