CS & IT ENGINEERING

Compiler Design

Lexical & Syntax Analysis

Lecture No. 1

By- DEVA Sir

01 Importance ?

02 Introduction

03 → Compiler ?

04 → phases ?

05 → Language Translation?

Q1) Do you require Any Prerequisite ?

↳ Ans: NO

But If you know C programming then you will enjoy.

- C programming basics
- Regular expressions basics
- CFG

Q2) Do you require any Text Book ?

↳ NO

If anyone would like to refer then Drgon edition

Q3) What is Weightage in GATE ?

5 – 7 Marks

Q4) How to prepare Compiler Design ?

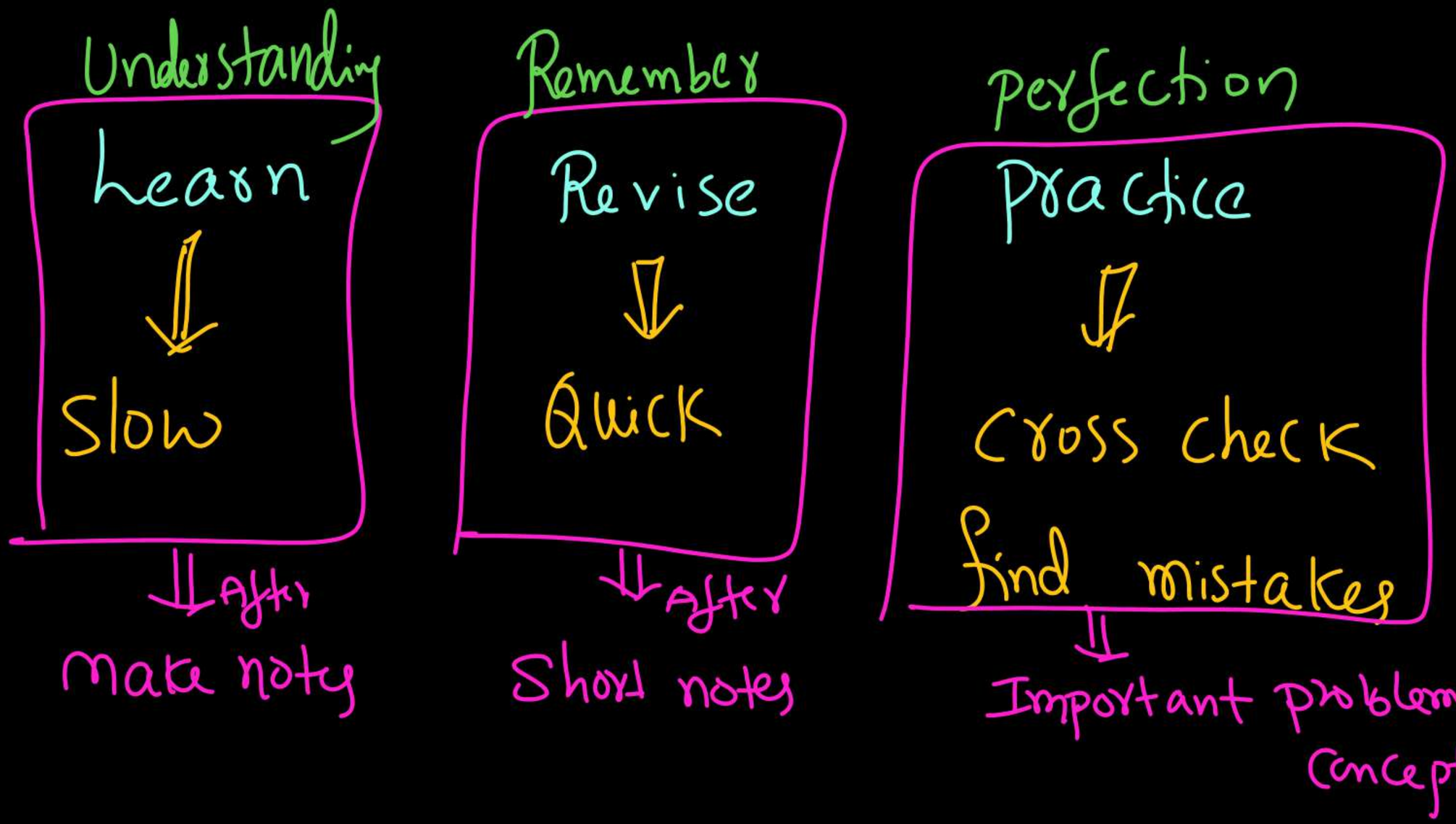
Systematic Approach :

1st : Attend class

2nd : Make the notes after class

3rd : Revision of notes

4th : practice class problems, Attempt DPPs and Solve GATE PYQs


Understanding
Learn
Slow
After
Make notes
Remember
Revise
Quick
After
Short notes
Perfection
Practice
Cross check
Find mistakes
Important problems/
Concepts

Short notes

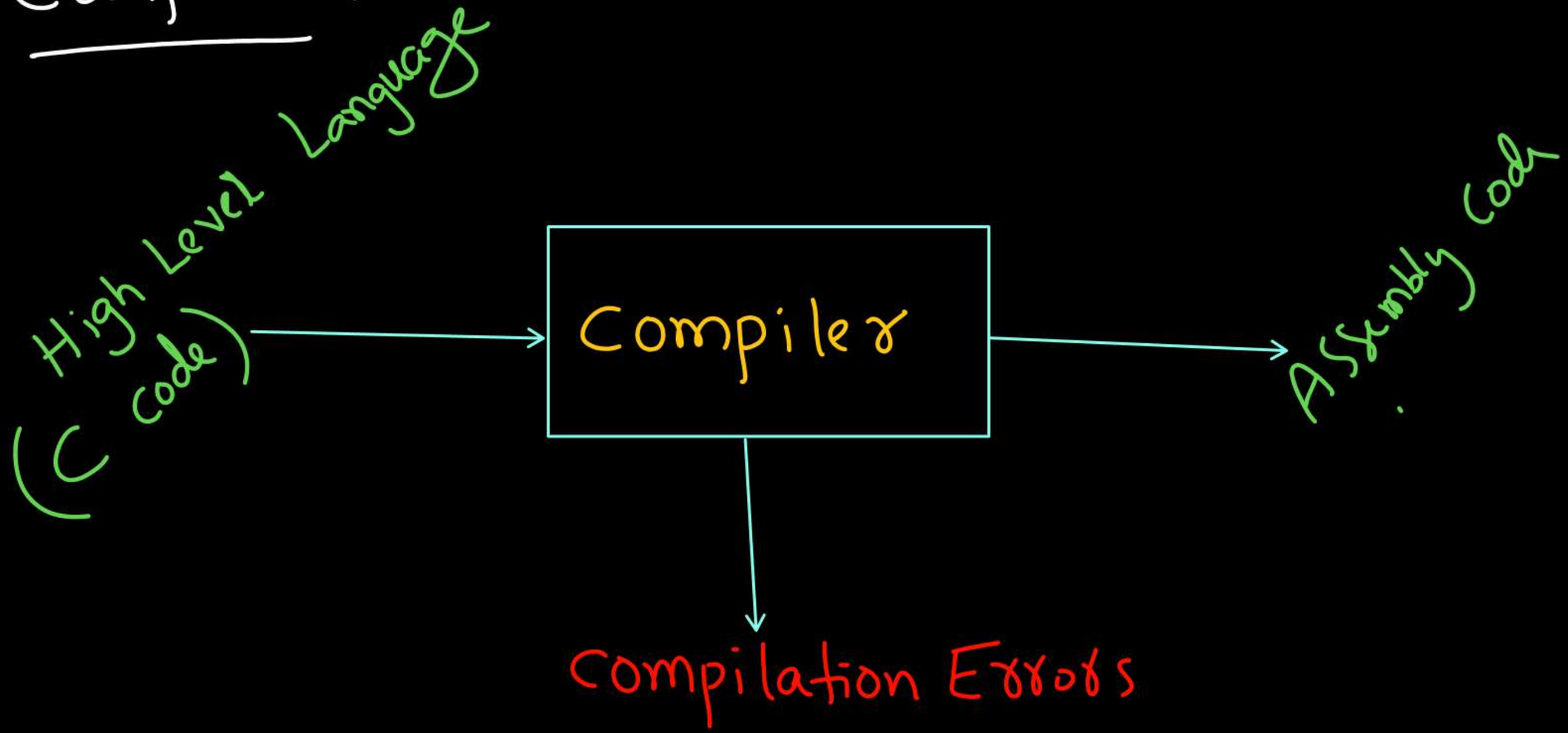
↳ what is Important ?

↳ only after completing topic/chapter/ Subject.

Compiler Design :

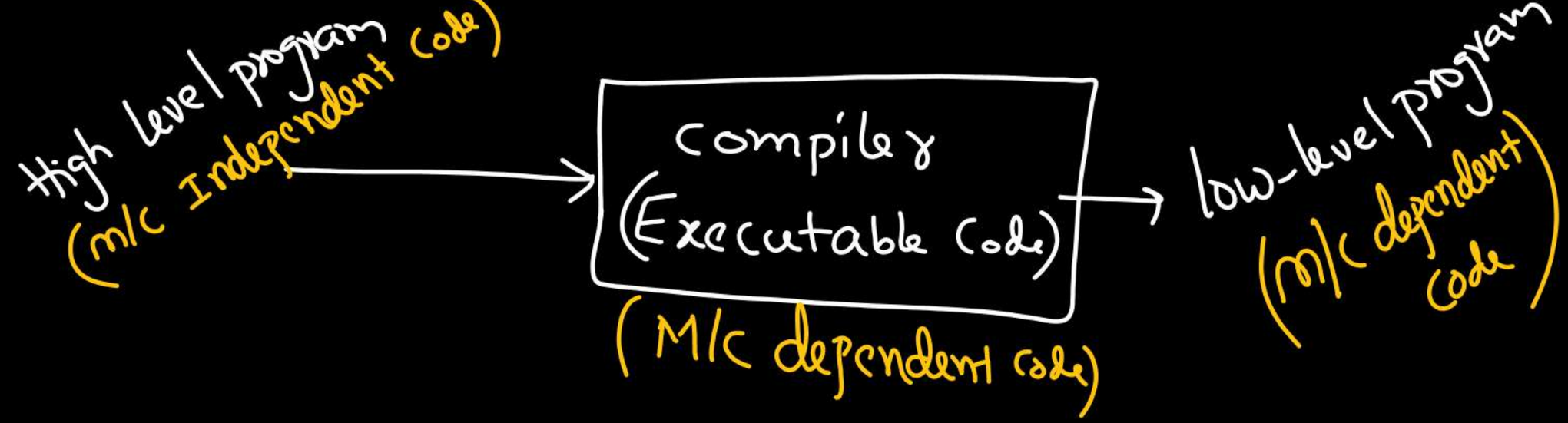
1. Introduction
2. Lexical Analysis
3. Syntax Analysis *** 3-4M
4. Syntax Directed Translations (SDTs)
5. Intermediate Code & Code optimization ***

Compiler :


High Level Language
(C code)
Compiler
Assembly Code
Compilation Errors

Compiler

- It is translator
- It is program
- It is executable



Portable

m/c Independent

High Level languages

Intermediate codes

Not portable

m/c Dependent

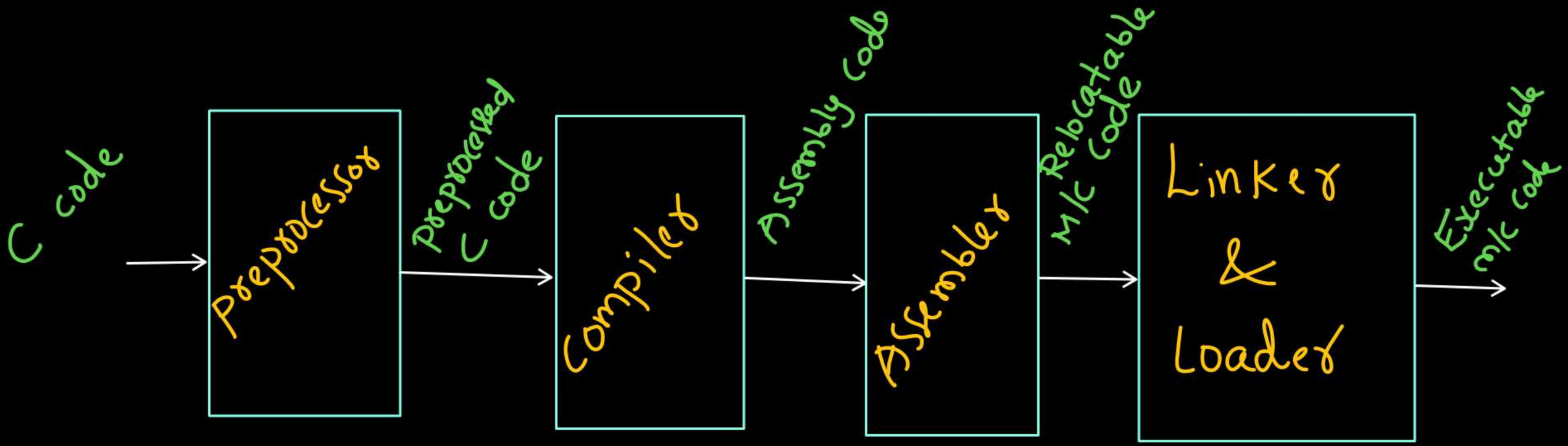
Compiler

Assembly code

M/C code

Translator

C Language Translation :

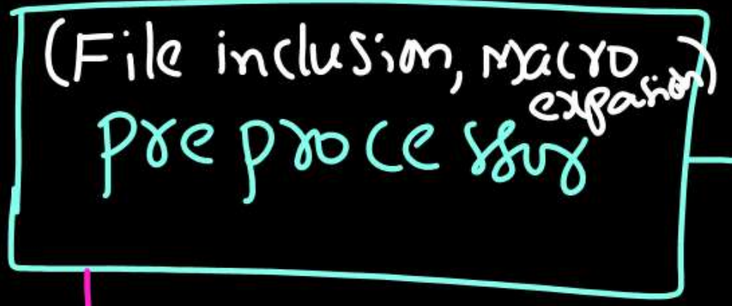


Preprocessor

#include <...>
#define <...>

#define MAX 10

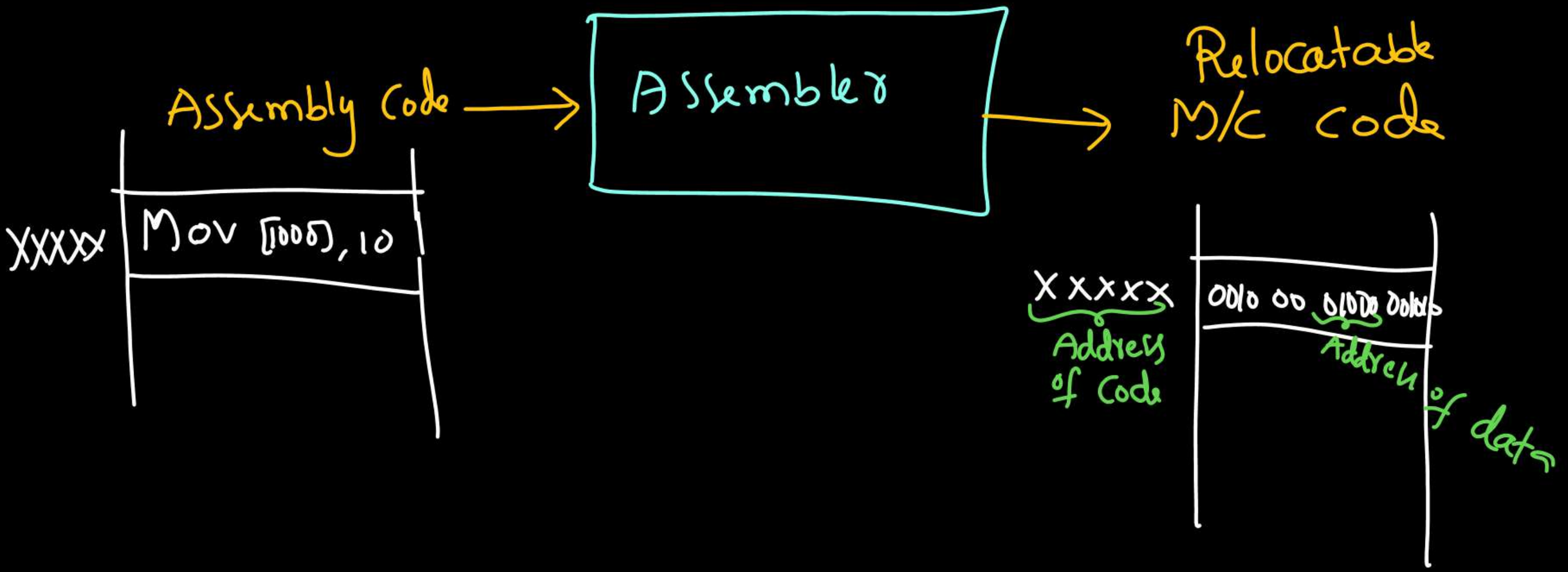
int a=MAX;



int a=10;

↳ Any statement begins with # is called as preprocessor statement

Assembler :


Assembly Code
Assembler
Relocatable M/C code
XXXX
Mov [1005], 10
XXXXX
Address of Code
Address of data

Linker :

↳ To resolve all external references

(all external files)
(all external variables & functions)

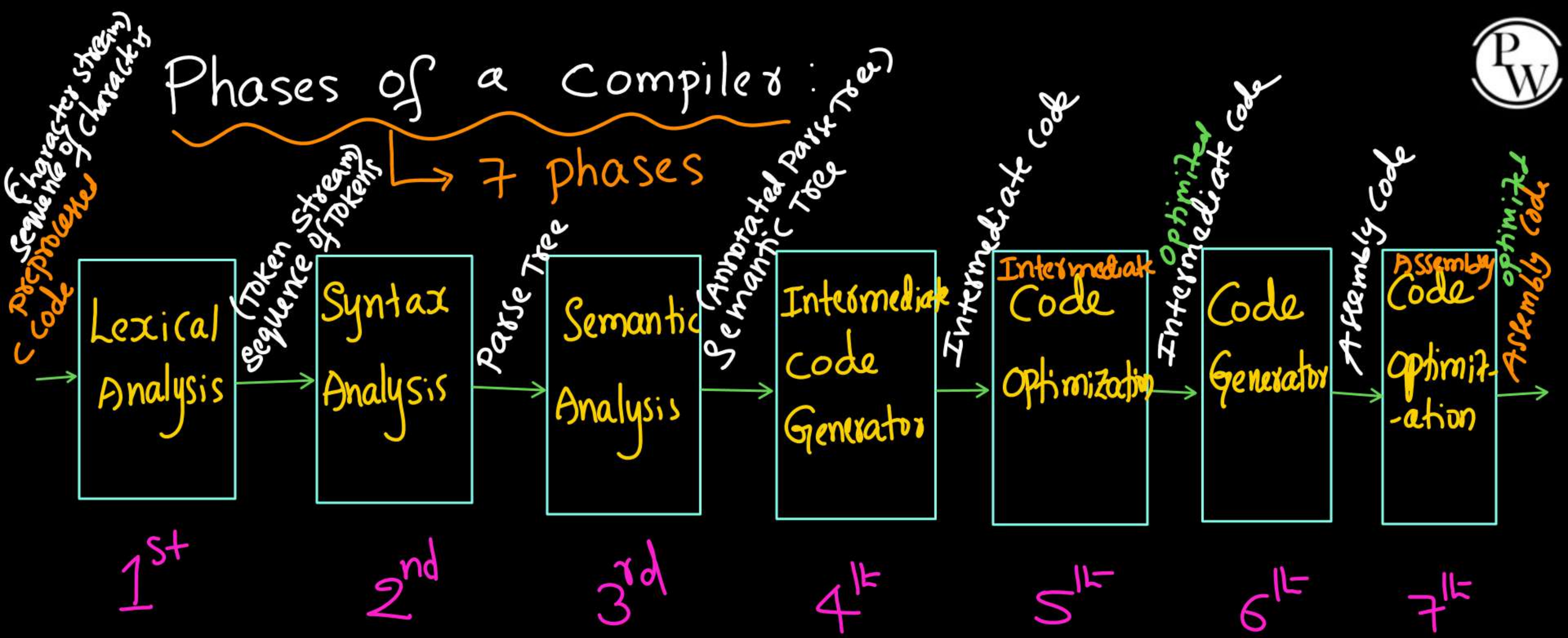
Loader :

↳ It performs "relocation".

↳ Altering address of code/data.

Phases of a Compiler :

↳ 7 phases



Match the following Groups :

Group-I [phase]

1. Lexical → A
2. Syntax → B
3. Intermediate Code Generator → D
4. Code Generator → E

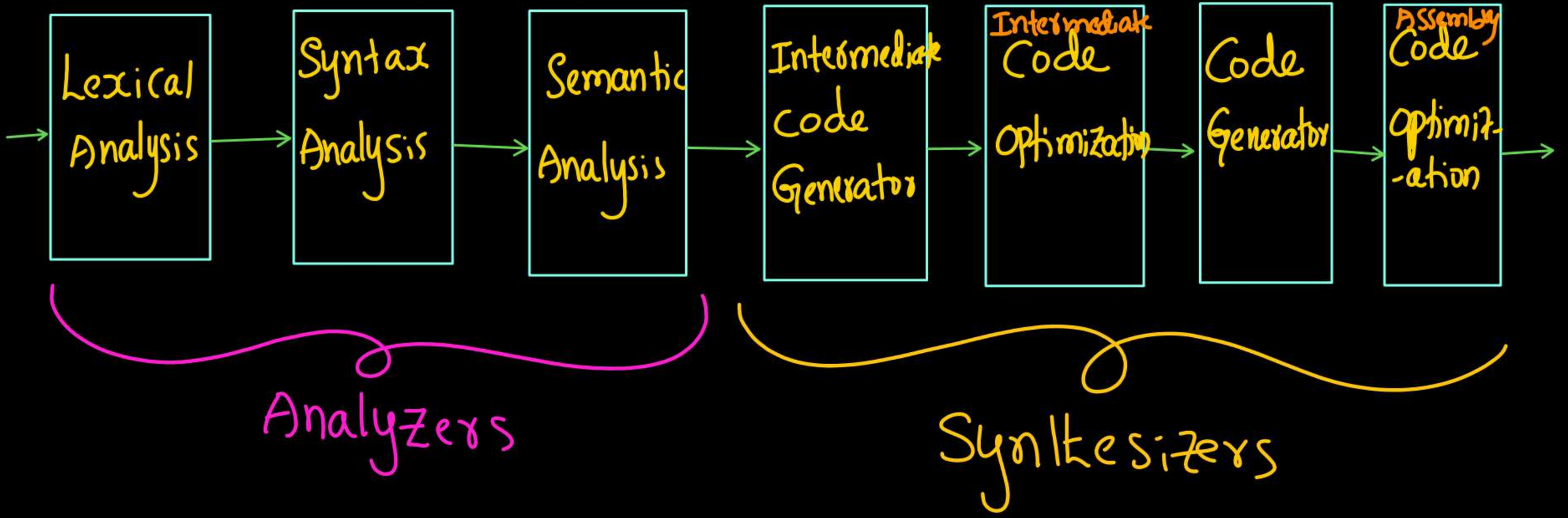
Group-II [Input]

A. Character stream
B. Token stream
C. Parse Tree
D. Annotated parse Tree
E. Intermediate Code
F. Assembly code

Match the following Groups :

Group - I [phase]	Group - II [Output]
1. Lexical → B	A. Character stream
2. Syntax → C	B. Token stream
3. Intermediate Code Generator → E	C. Parse Tree
4. Code Generator → F	D. Annotated parse tree
	E. Intermediate code
	F. Assembly code

Phases of a Compiler :



You	am	Student
I	are	Students
(I)	(am)	(Student)

Lexical	Syntax	Semantic
I	S + V + Obj	Meaning
Words	Structure	
am		
Student		

y = 20 ;
x = 10 ;

x = 10 ;
integer integer

Lexical	Syntax	Semantic
x ✓	id = constant ;	x type is Same as 10 type
= ✓		
10 ✓		
; ✓		

C code

I code

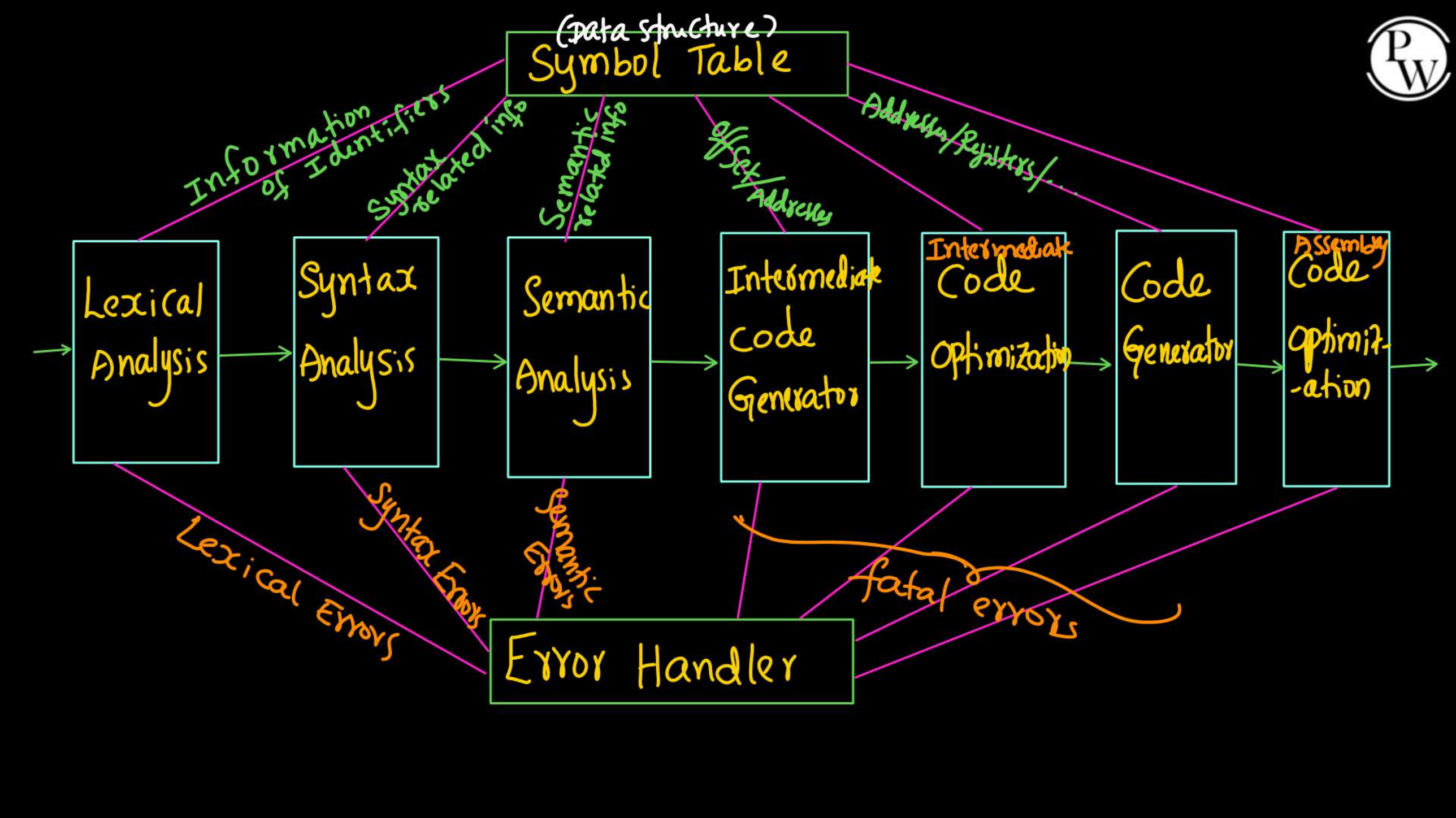
} M/c Independent

A code

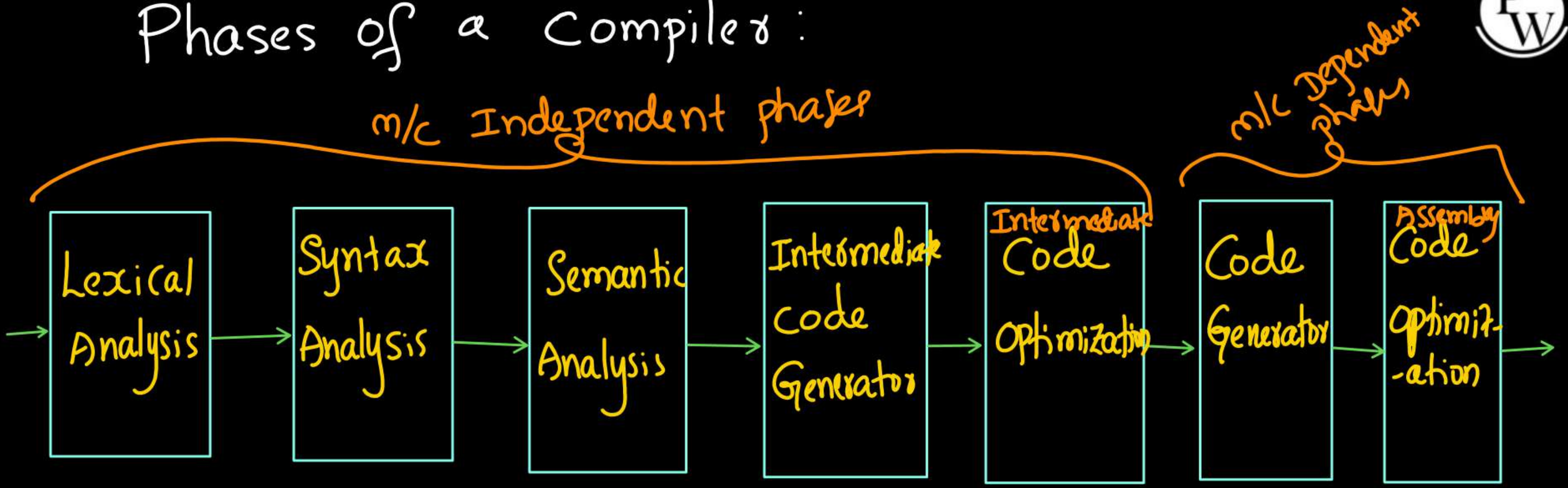
M Code

E. code

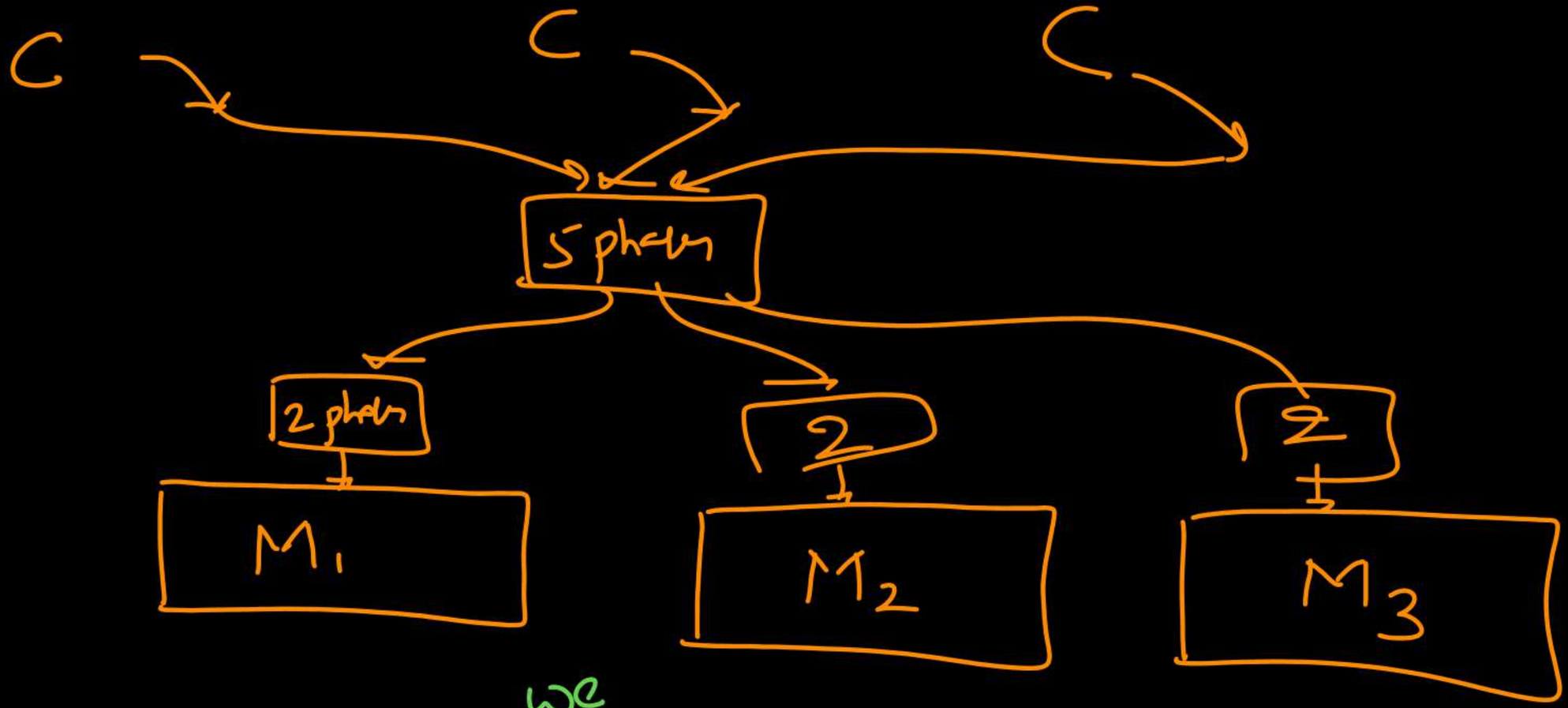
} M/c dependent


(Data structure)
Symbol Table
Information of Identifiers
Syntax related info
Semantic related info
offset/Address
Addresses/Registers/...
Lexical Analysis
Syntax Analysis
Semantic Analysis
Intermediate code Generator
Intermediate Code Optimization
Code Generator
Assembly Code optimiz-ation
Lexical Errors
Syntax Errors
Semantic Errors
fatal errors
Error Handler

Phases of a Compiler :

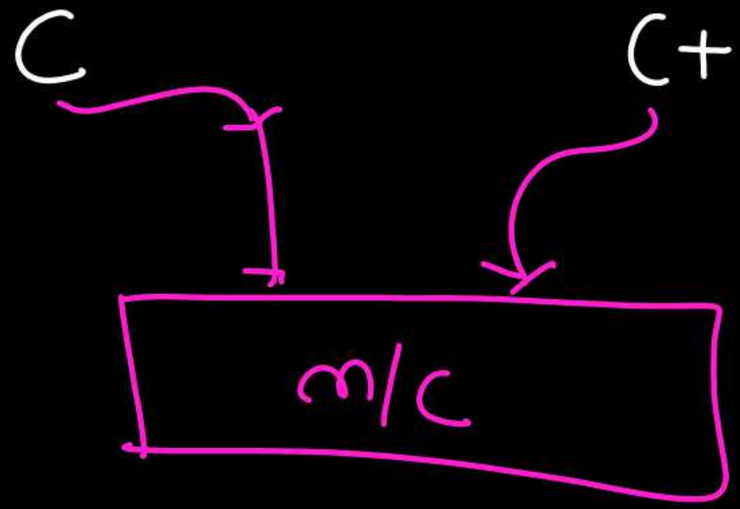


Cost of one Compiler = 7



If we don't optimize, 21 phases

Same C program, if we want to run on 3 different machines, How many phases? = 21.



If 2 different programming languages need to be Compiled on Same m/c,

How many phases ?

Summary

- Compiler
- Language Translation
- Preprocessor
- Assembler
- Linker
- loader

Symbol Table

Lexical Analysis

Errors


THANK YOU
GW
SOLDIERS !