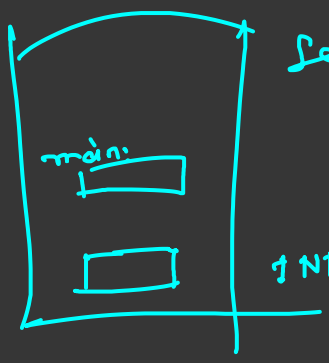
ICG

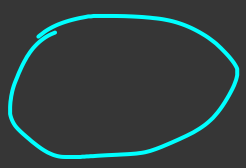
C → 8086 Asm (Runnable)

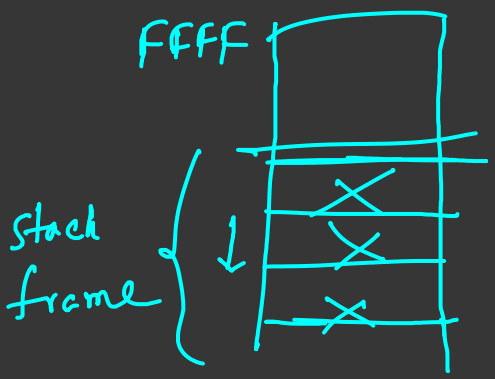


BIOS

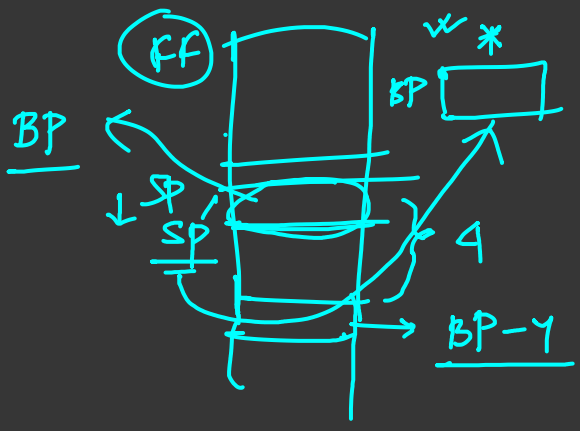
f( ){

}



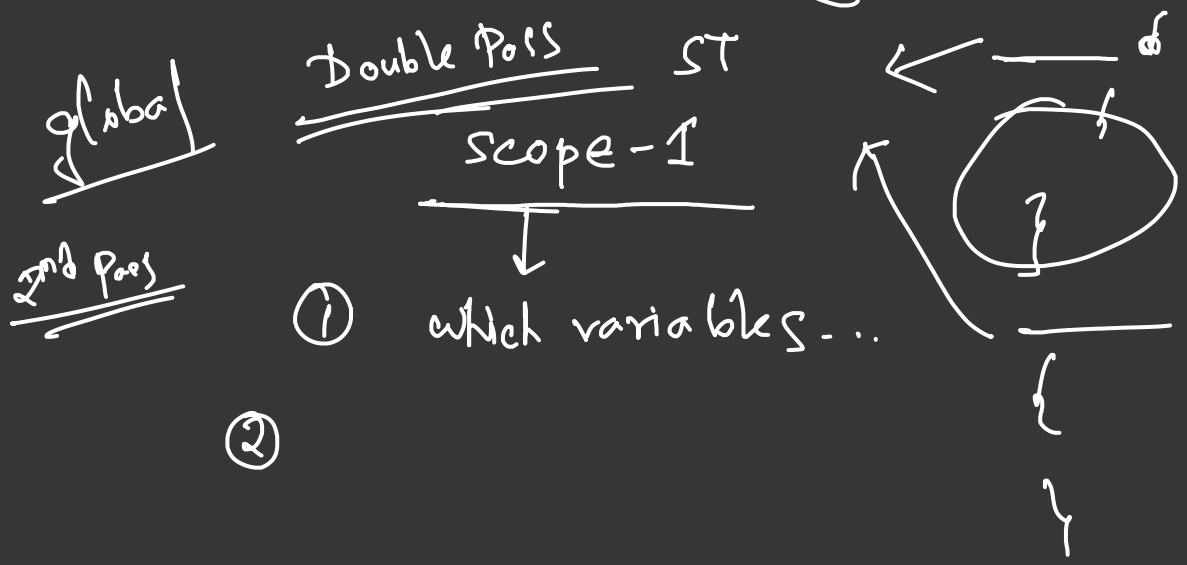



```
f PROC
    push BP
    MOV  BP, SP
    .
    .
    pop BP
    RET 2*(arg-count)
f ENDP
```

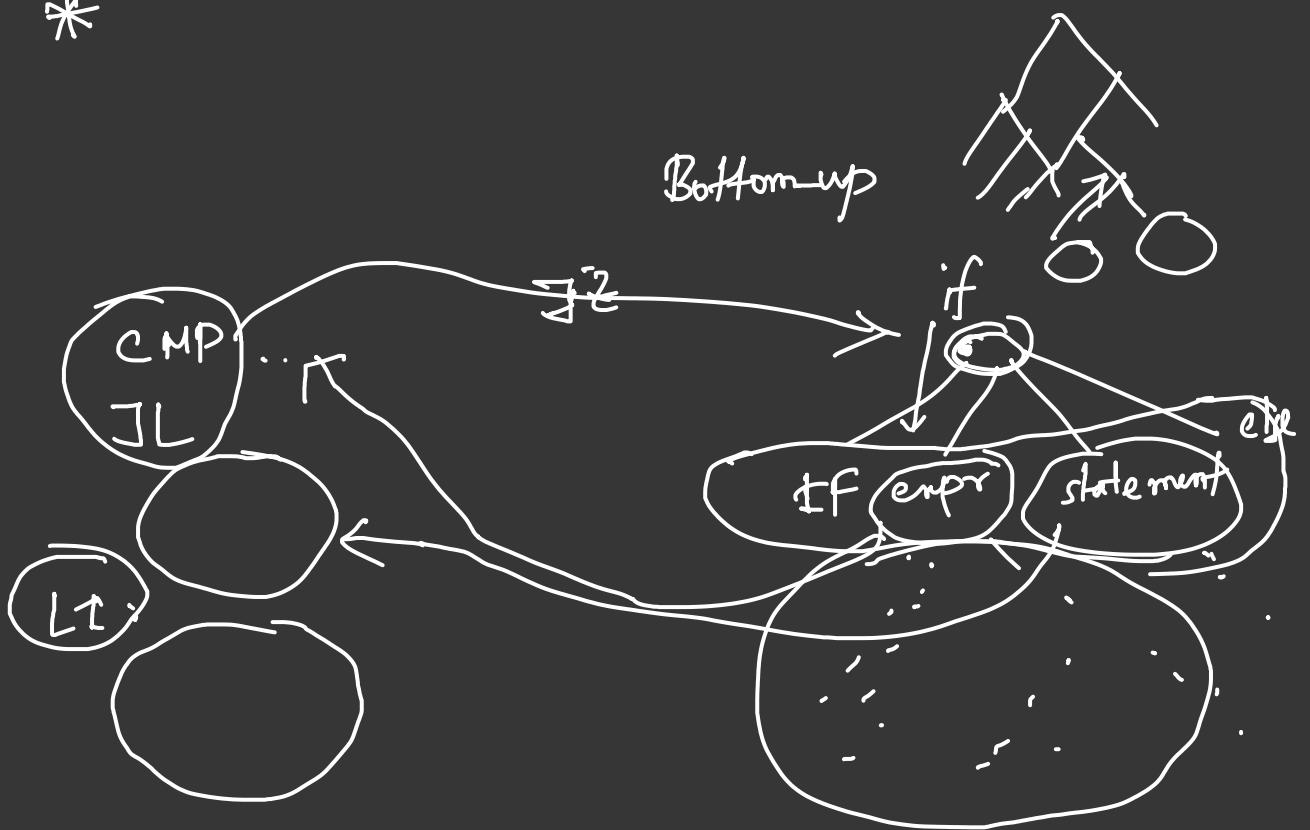


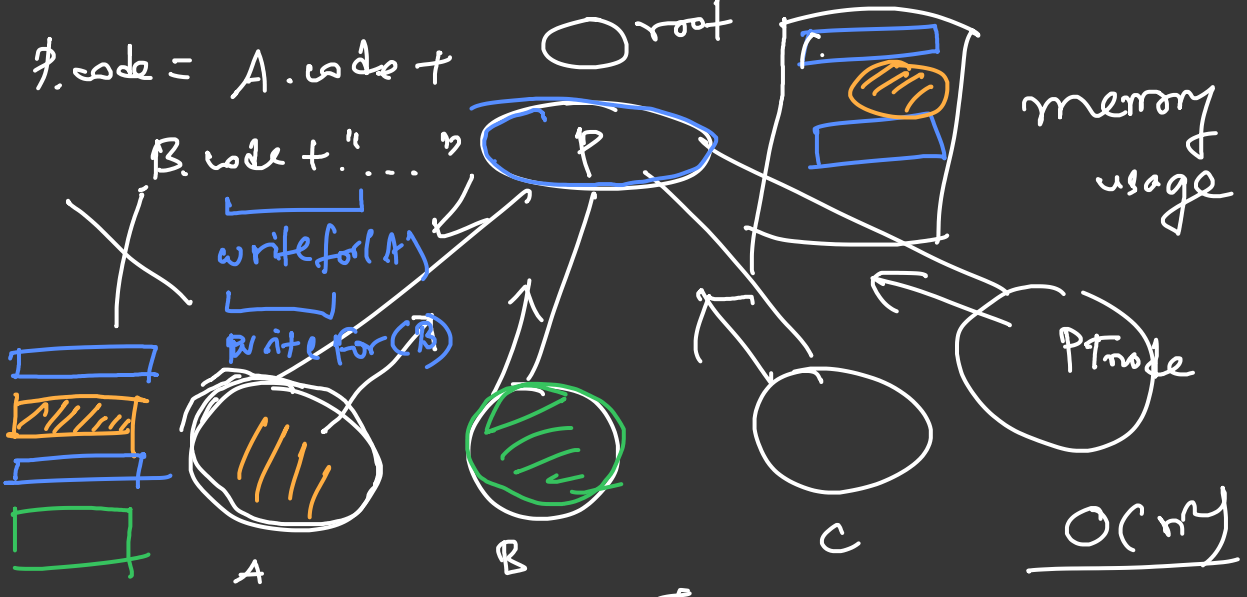
function — argument
— return

Variables — local
— global } .....



*





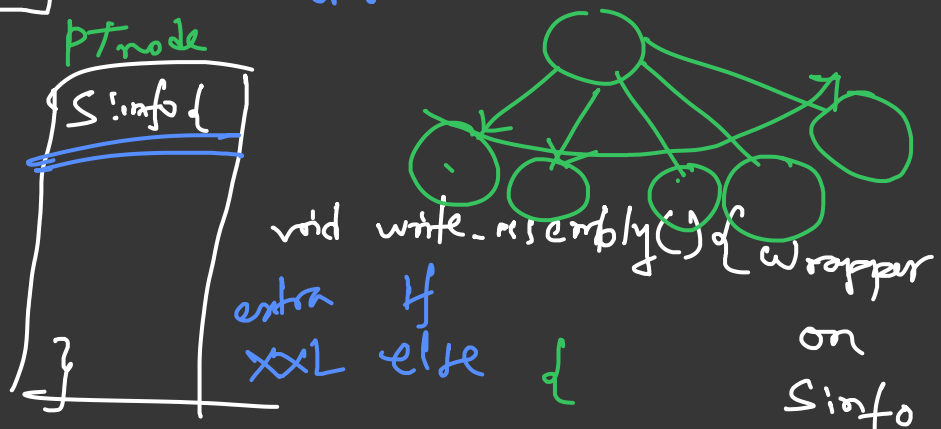
ParseTree — for each non terminal

① RecursiveDescent / ...

② Top down — when I am at P → I will keep adding my appendage and child code order

③ PostOrder(root)

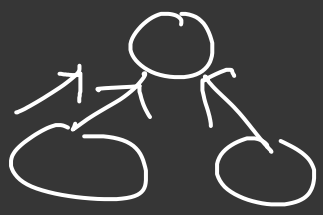
✓



on-the-fly

mode → Output

Double-Pass



local variable

→ .DATA ✗

stack

↖ .DATA

a .....

b .....

c . .

b = 3

int

a, b, c[5], d[10] ;

2

FF

[BP-4]

MOV [BP-4], 3

SUB SP, 2

SUB SP, 10

5

SP

BP

SP

3

globalCount

ADD SP, cnt

main

mov BP

RET

a = 5

MOV [BP-2], 5 ✓

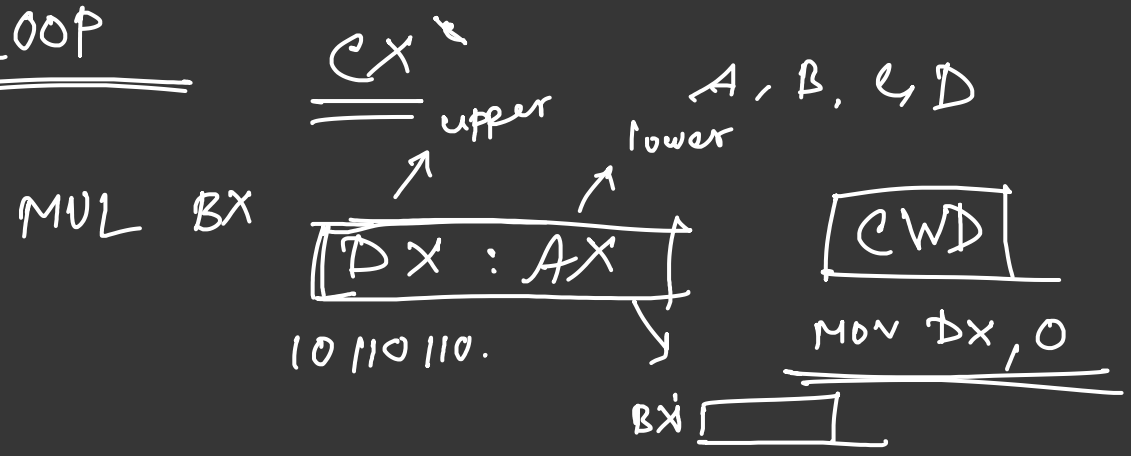
1st Pass

0 0

4 ⟶ stack offset

2nd pass

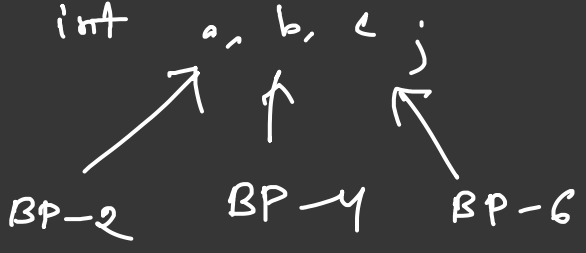
LOOP



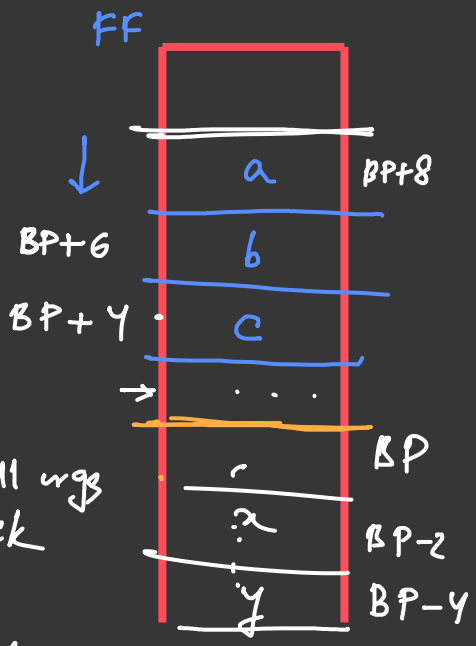
DX → Remainder

AX → Quotient

D Local Var




```
int f(int a, int b, int c) {
    int x, y;
    x = 2; // MOV [BP-2], 2
    a = 4; // MOV [BP+8], 4
}
```



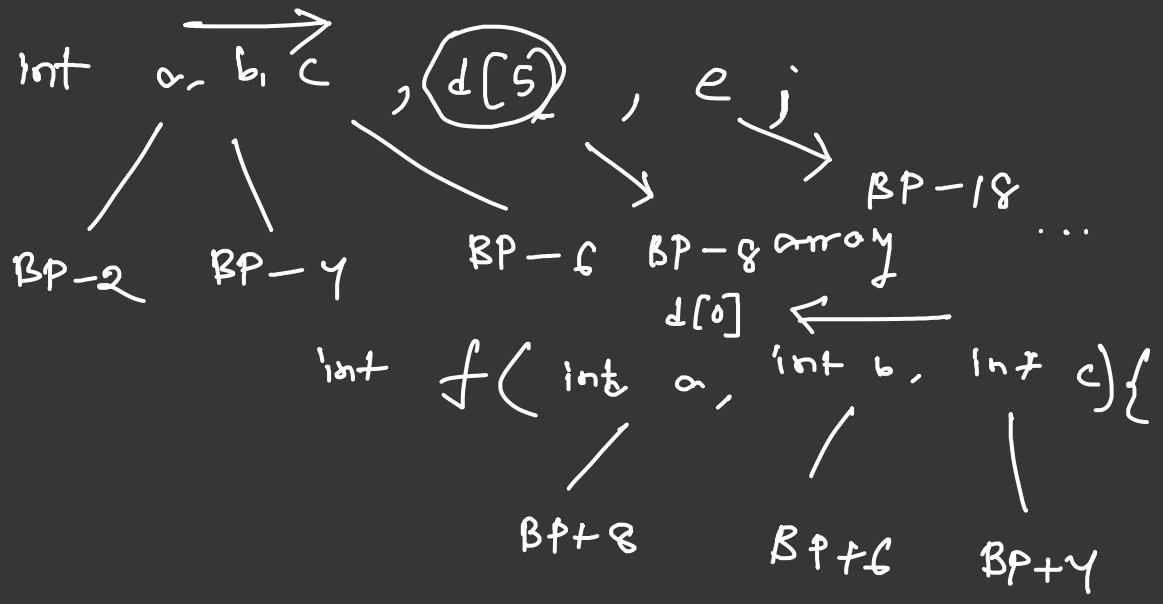
Caller

{ push all args to stack

CALL f

f(1, 2, 3)

```
MOV AX, 1
PUSH AX
MOV AX, 2
PUSH AX
  :
CALL f
```



return

3 {

intf.. {

if(. .) {

}

return

MOV AX,..

JMP ret_label

}

return

CALL f

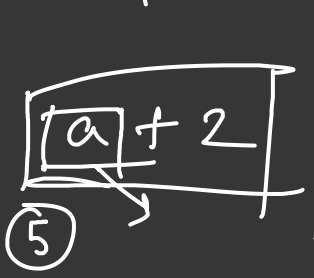
AX . . .

ret_label:

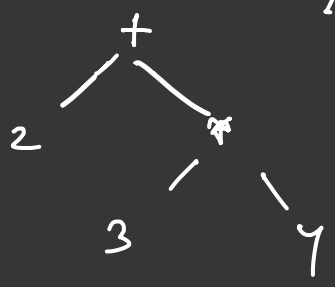
P → Q

Expression

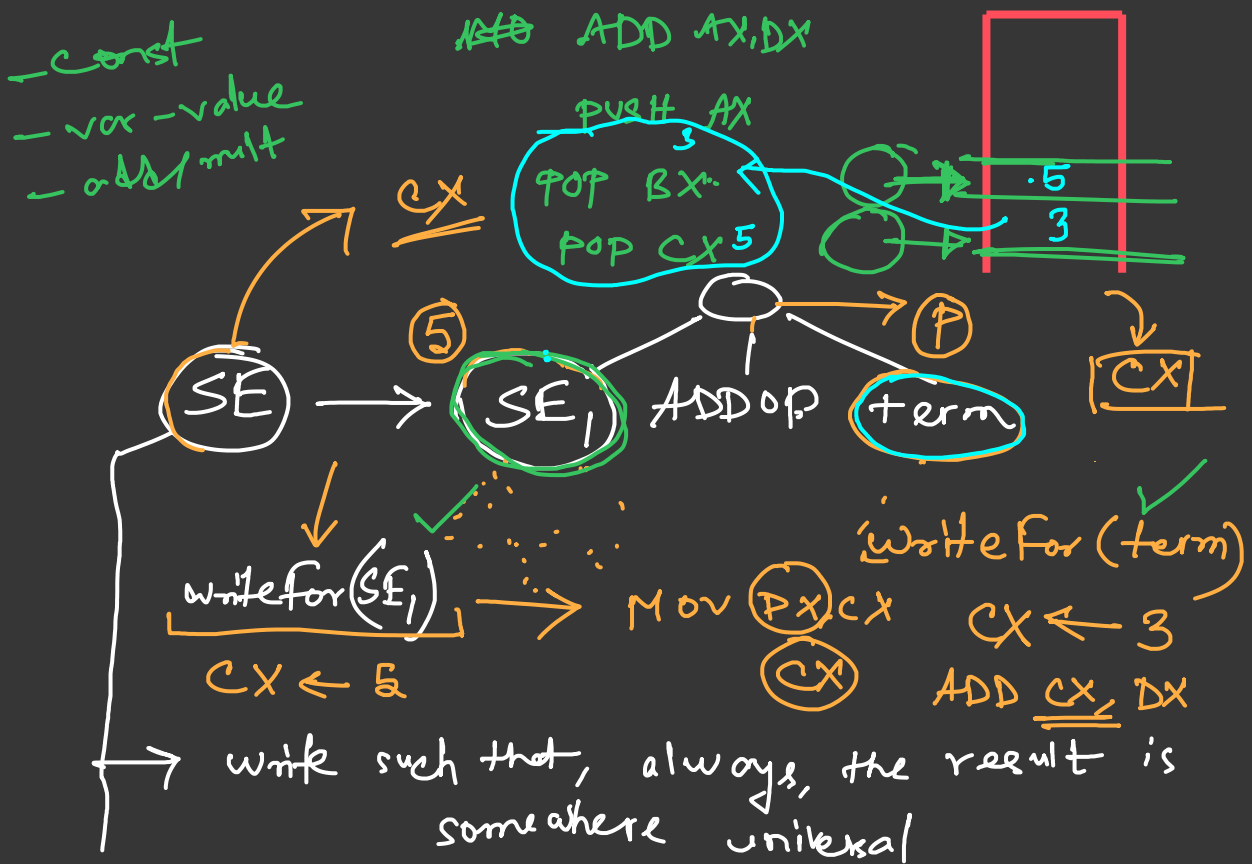
2+3*4



AST



(*) Whenever, a "new value is prepared", push to stack



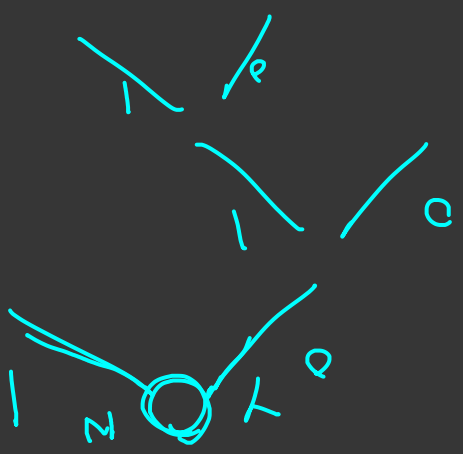
Approaches ① write result in CX or, BX

② stack push

If, Loop, Array

Hashmap

Thank You <3


H
C
O
N
O