ASSIGNMENT 1

1. Perform macro expansion on the following code:

#define square (x) ((x) \* (x))

main( )

{

int num = square(5);

}

ANS:-

main() main()

{ {

Int num=(5\*5); ====🡺 int num=25;

} }

1. Discuss action taken by every phase of compiler on the following string:

Sum = OldSum – Value /100

Ans:

* **Lexical-Analyzer:**

In this phase for this Expression it creates stream of tokens,

Id1=id2 - id3 / 10 ;

Tokens:

Sum -------- identifier

= -------- Assignment Operator

OldSum ---------- identifier

* --------- Artihmatic operator

Value -------- Identifier

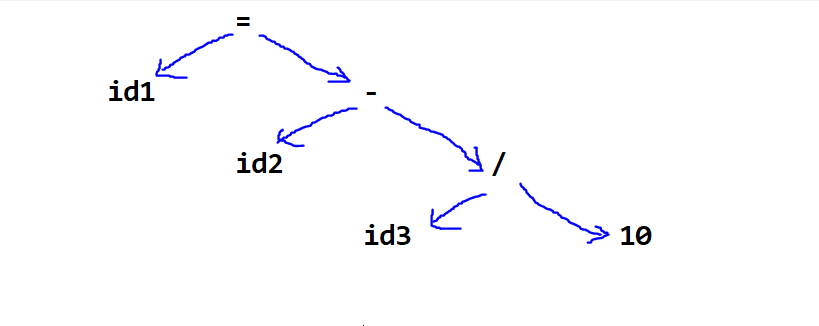
/ -------- Arithmatic Operator

10 ---------- NumericConstant

* **Syntax Analyzer**

**Creates Parse tree for ,**

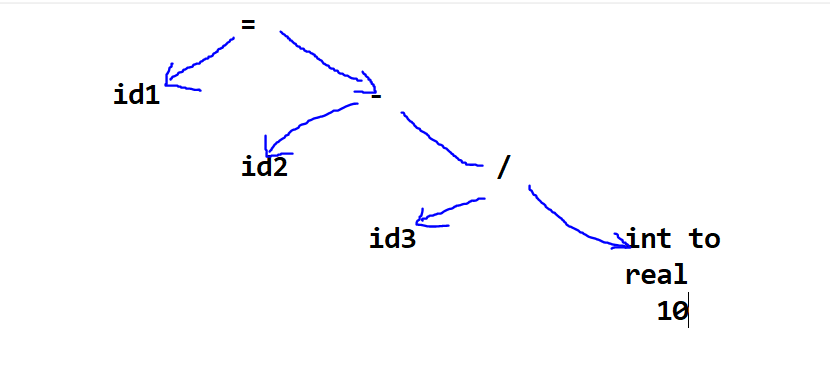
**Id1= id2 – id3 / 10 ;**



* **Symentic Analyzer :**

**Creates Symentically verified parse Tree ,**

**Lets suppose id1 , id2 & id3 all are in Float and ‘10’ is in Int, so for that…**



* **Intermediate Code Generator :**

**Creates intermediate code using three address space,…**

**T1 := int to real ( 10 )**

**T2 := id3 / T1**

**T3 := id2**

**T4 := T3 – T2**

**Id1 := T4**

**Code Optimization**

**Optimizes code…**

**T1 := id3 / 10.0**

**Id1 := id2 – T1**

## Code Generation

**MOVEF id3 , R1**

**DIVF R1 ,10.0**

**MOVEF id2, R2**

**SUBF R2 , R1**

**MOVEF R2 , id1**

1. Identify and eliminate useless symbol from the following grammar:

