Q) Show the o/p of each phase of compilation

```
int n = 0;
    for (i=0 ; i<5 . i++)
    {
        n= n+ 50;
    }
```

A] Lexical Analysis

Lexeme	Token
int	< keyword >
n	< id,1 >
=	< =, op >
0	< Num,0 >
for	< keyword >
i	< operator >
i	< identifier >
=	op

< No, 0 > <;> < id,2> << , reloop >.
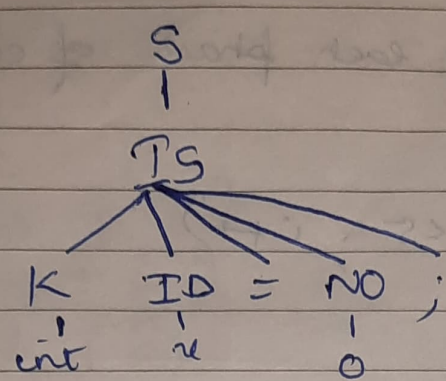< NO,5 > <;> < id,2> <++> <)>

3] < { >

4] < id, 1 > < = > < id,1 > < + > < 5.0 > <;>

5] < 3 >

B] Syntax analysis
$$S \rightarrow IS$$
$$IS \rightarrow K \quad Id = NO;$$

```
                    S
                    |
                   TS
                  ╱↑｜＼＼
                 K   ID = NO ;
                 ┊   ┊    ┊
                int   x    0
```

$S \rightarrow IS \mid FS \mid AS$

$IS \rightarrow K \quad id = NO.$
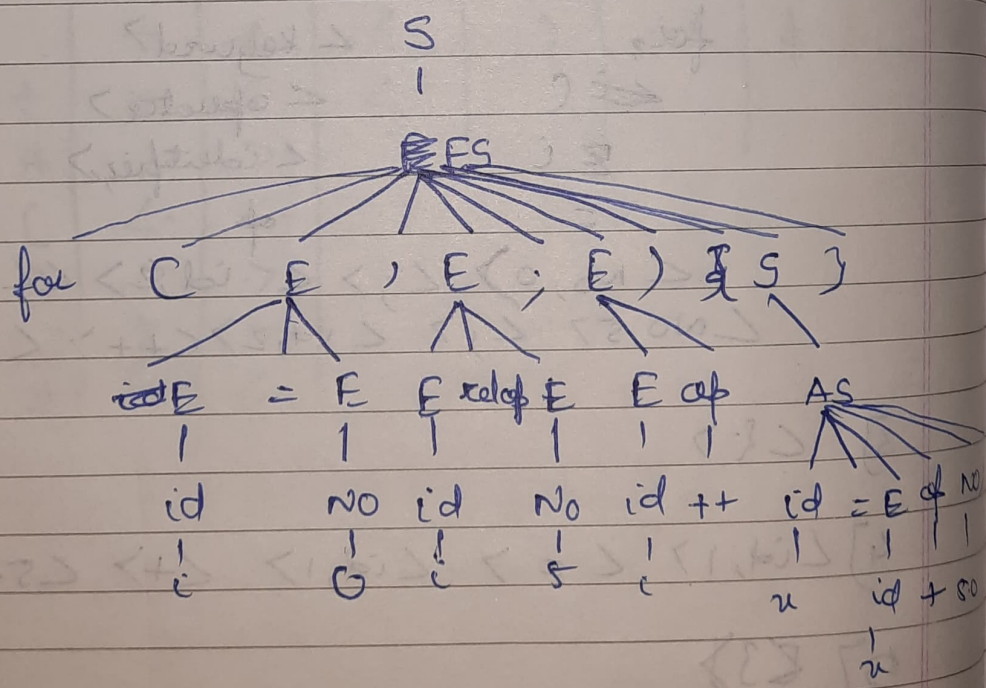
$FS \rightarrow for (E, E, E) \{S\}$

$AS \rightarrow id = E \text{ op } NO.$

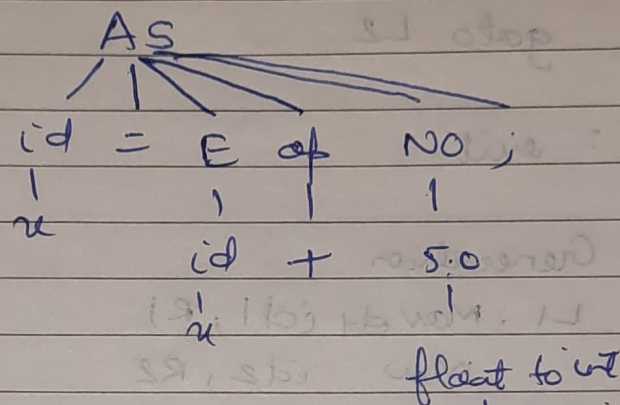$E \rightarrow E = E \mid E \text{ relop } E \mid E \text{ op} \mid id \mid No.$

$op \rightarrow + \mid ++$

$relop \rightarrow <$

```
                          S
                          |
                         FS
              ╱╱╱╱↗↑＼＼＼＼＼
         for < C >  E < , E ; E ) { S }
                   ╱┃＼   ╱┃＼  ╱＼   ╱＼
              idE  =  E  E relop E  E op   AS
               |     |   |     |  | |     ╱↑＼
              id     No  id    No id ++   id = E op NO
               ┊     ┊   ┊     ┊  ┊       ┊  ┊   ┊
               i     O   i     S  i       x    id + 50
                                                   ┊
                                                   x
```

C) Semantic Analysis (Type checking)

AS

id = E of NO

x

id + 5.0

x

float to int

D) Intermediate code Generation

L1 : id1 = 0

temp 1 = float to int (5.0)

id2 = 0

L2 : if (id2 < 5) goto L3

goto L4

L3 : id1 = id1 = id1 + temp1;

id2 = id2 +1

goto L2

L4 : exit

E) Code optimization

L1: id1 = #0

id2 = #0

L2 : if (id2 >= #5) goto L4

L3 : id1 = id1 + temp1,
   id2 = id2 + #1
   goto L2

L4 : exit.

F) Code Generation
   L1 : Mov id1, R1
        Mov   id2, R2
   L2 : CMP   R2 , #5
        JGTE L3
        ADD   #5, R1
        ADD   #1, R2
        JMP   L2
   L3 : MOV   R2, id2
        MOV   R1, id1
        EXIT.

S → ABC
A → bAS/ε
B → cASB/ε
C → AB

r_i	First (x)
S	b c E
A	b E
B	c E
c	b E C