Assignment -1

Q1. Follow below steps and implement code in C for scanner.

Step 1	Generate text file for given Input
Step 2	Declare two static table for Operator and Keywords
Step 3	Declare two dynamic table for constant and Symbol
Step 4	Read Input file apply STRTOK () to tokenize given input string (get logic from Help menu)
Step 5	In tokenization While loop, for each token • Check for keywords from keyword table if it exists then print [KW#index], where index is the record number in the respective table. • Else Check for Operator from Operator table if it exists then print [OP#index], where index is the record number in the respective table. • Else check that given token is digit then check whether it exists in constant table then print [CO#index], where index is the record number in the respective table, else store digits in constant table then print [CO#index] • Else check that given token exists in symbol table then print [ID#index], where index is the record number in the respective table, else store symbol in symbol table then print [ID#index]
Ex:	INPUT : INT a , b ; REAL c , d ; a = b + c * 100 ; d = a – 90 ;

Static Table:

OP TABLE	,	;	=	*	+	-

KW TABLE	INT	REAL

Dynamic table:

ID TABLE	a	b	c	d

CO TABLE	100	90

OUTPUT :

```
[KW #1] [ID #1] [OP #1] [ID#2][OP#2]
[KW#2] [ID#3][OP#1][ID#4][OP#2]
[ID#1][OP#3][[ID#2][OP#5][ID#3][OP#4][CO#1][OP#2]
[ID#4][OP#3][ID#1][OP#6][CO#2][OP#2]
```

Q2. Follow below step and implement code in c for DFA.

Step 1	Declare STT for int, real and id
Step 2	Declare s_table[6][4][6]

State	d	l	.
Start	INT	ID	Error
INT	INT	Error	S
S	REAL	Error	Error
REAL	REAL	Error	Error
ID	ID	ID	Error

Step 3	Take input; pre_state = ”start”
Step 4	Calculate length of input = len
Step 5	Int k=0; c=0; i=0;

Step 6	While (k<len)
	{
	for i=1 to 6{
	compare pre_state with s_table[i][0] & decide row of s_table;
	find out column “c” of the s_table by calling function for checking character is letter, digit or dot(.)
	copy s_table[i][c] to cur_state;
	break;
	}
	Copy pre_state = cur_state
	K++;
	}

Step 7	Print (pre_state, current character, cur_state)
Ex.	INPUT: 123 OUTPUT: START 1 INT INT 2 INT INT 3 INT Valid Integer

Exercise---

1) Implement DFA for INT.
2) Implement DFA for ID.

Q3. Implement RD parser for checking input string is valid or not and print parse tree in postorder.