

List of Experiments

S. No.	Topic(s)	CO	PO
1	A. Write a simple calculator program in C/C++/JAVA B. Implementation of basic Flex programs	1	1
2	Implementation of Lexical Analyzer using FLEX.	1	2,5
3	Implementation of calculator using FLEX and BISON.	1	2,5
4	Write a program for Left recursion/Left factoring in C/C++/JAVA	2	2,5
5	Write a program for to Compute FIRST & FOLLOW for Top-Down Parsing and predictive parsing table in C/C++/JAVA	2	2
6	Write a program for Shift Reduce Parsing in C/C++/JAVA	3	2
7	Write a program for Computation of LEADING AND TRAILING in C/C++/JAVA	3	2
8	Write a program for Computation of LR (0) items in C/C++/JAVA	3	2
9	Write an program for Intermediate code generation as Prefix and Suffix in C/C++/JAVA	4	3
10	Write an program for Intermediate code generation as Quadruple, Triple, Indirect triple in C/C++/JAVA	4	3
11	Write a program to generate machine code for a simple statement in C/C++/JAVA	5	3
12	Implement backpatching in C/C++/JAVA	5	3

Write a program for Left recursion/Left factoring in C/C++/JAVA **Procedure:**

Input: Grammar G

Output: Parsing table

1. left recursion from the grammar G.

For each non-terminal A in G, do the following:

If there exists a production $A \rightarrow A\alpha \mid \beta$, where β is not starting with A, split it into:

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A' \mid \epsilon$ (epsilon), where ϵ represents the empty string.

If $A \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$ are the productions of A after step 1, remove β_i if β_i starts with A.

2. left factoring from the grammar G

For each non-terminal A in G, do the following:

If there exists a production $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$ where A is a non terminal and β

$A \rightarrow \alpha A'$

$A' \rightarrow \beta_1 \mid \beta_2$

TASK TO BE GIVEN TO THE STUDENTS - KINDLY GIVE THE INPUT CASES DIFFERENTLY FOR EACH STUDENT -REFER BELOW

LEFT FACTORING INPUTS

1. $S \rightarrow abS \mid aSb$

2. $S \rightarrow aSB \mid aBb \mid aSb$

3.
$$\begin{aligned} S &\rightarrow aCd \mid aT \\ C &\rightarrow a \mid ab \\ T &\rightarrow ccd \mid ddc \end{aligned}$$

4. $A \rightarrow bAAaaA \mid bAAaAb \mid bAc \mid a$

5.
$$\begin{aligned} S &\rightarrow iEtS \mid iEtSeS \mid a \\ E &\rightarrow b \end{aligned}$$

6. $A \rightarrow aAB \mid aBc \mid aAc$

7. $S \rightarrow aSSbS \mid aSaSb \mid abb \mid b$

8. $S \rightarrow a \mid ab \mid abc \mid abcd$

9.

```
S → aAd / aB
A → a / ab
B → ccd / ddc
```

LEFT RECURSION

1. $S \rightarrow S + S \mid S * S \mid a$

2. $A \rightarrow ABd \mid Aa \mid a$
 $B \rightarrow Be \mid b$

3. $S \rightarrow (L) / a$
 $L \rightarrow L, S / S$

4. $S \rightarrow Sa / \varepsilon / bB / bD$
 $B \rightarrow b$
 $D \rightarrow d$

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

5. $F \rightarrow (E) \mid id$

$S \rightarrow a \mid (T)$

6. $T \rightarrow T, S \mid S$

7. $S \rightarrow SS+ \mid SS* \mid a$

8. $S \rightarrow S0s1s \mid 01$

