OR

(b) Calculate First and FOLLOW of given grammar. Design predictive parsing table : (CO1, CO2)

$S \rightarrow A)$

$A \rightarrow A, P) (P, P$

$P \rightarrow \{num, num\}$

H Roll No. .....

TCS–601

B. TECH. (CSE) (SIXTH SEMESTER)
MID SEMESTER
EXAMINATION, April, 2023

COMPILER DESIGN

Time : 1½ Hours

Maximum Marks : 50

Note : (i) Answer all the questions by choosing any ***one*** of the sub-questions.

(ii) Each sub-question carries 10 marks.

1. (a) What is compiler ? Describe how various phases could be combined as a pass in a compiler. (CO1, CO2)

OR

(b) Consider the following grammar :

$S \rightarrow ABC$, $A \rightarrow Aa \mid d$, $B \rightarrow Bb \mid e$, $C \rightarrow Ce \mid f$

Eliminate left recursion from the above grammar. (CO1, CO2)

P. T. O.

2. (a) Explain Token, Pattern, Lexemes and recognize tokens for given code : (CO1, CO2)

```
int MAX (int a, int b) {
If (a < b)
Return a;
else
return b;
```

OR

(b) Write short notes on the following : (CO1, CO2)

(i) Lexical Analyzer

(ii) Left Factoring

(iii) Lex compiler

(iv) Symbol Table

3. (a) Find the NFA recognizing the language described by the following regular expression : $(a + b) * a (a + b)$. Then convert it into a equivalent DFA. (CO1, CO2)

OR

(b) Test whether the grammar is LL(1) or not and construct a predictive parsing table for it : (CO1, CO2)

S–AaAb | BaBa, A→e, B→e

4. (a) Design SLR(1) the following grammar : (CO1, CO2)

S → aAb|bB

A → Aa|ε

B → Bb|ε

OR

(b) Consider the following grammar : (CO1, CO2)

S → aAd | bBd | aBc | bAc

A → e

B → e

Construct LALR.

5. (a) Construct SLR parsing table for given grammar and implement parsing table for given string : (CO1, CO2)

"id * id + id"

E → E+T|T

T → T*F|F

F → (E)|id