

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 10457

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2025.

Sixth Semester

Computer Science and Engineering

CS 8602 — COMPILER DESIGN

(Common to : Computer Science and Business Systems)

(Regulations 2017)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Compare and contrast compiler and interpreter.
2. List the different phases in compilation.
3. Write a regular expression representing the strings whose length is divisible by 2 for $\Sigma = \{a,b\}$.
4. Write the reason for reduce-reduce conflict occurrence in CLR.
5. Write the context free grammar for representing palindrome of odd-length.
6. State Left recursive and Left factoring with an example.
7. What is activation record?
8. How to access the non-local data?
9. Mention the characteristics of Loop optimization.
10. What is the cost of the following instruction?
ADD *R1, R2[0]
SUB R3, #3
MOV M, R0.

PART B — (5 × 13 = 65 marks)

11. (a) (i) Explain the process of input buffering. How is it implemented for processing the token separation? (6)
- (ii) List the various types of token formats. Write regular expression for representing each of the token format. (7)

Or

- (b) Construct DFA for the regular expression $(ab) + (a+b)^*ba$ using Thompson's method and by subset construction.
12. (a) Consider the following grammar
- $P \rightarrow SbP \mid SbS \mid S$
 $S \rightarrow WbS \mid W$
 $W \rightarrow L^*W \mid L$
 $L \rightarrow id$
- (i) Construct predictive parsing table. (5)
- (ii) Is the grammar an LL grammar? (2)
- (iii) Check whether the string 'id * id b id * id' will be parsed by predicative parser. (6)

Or

- (b) Consider the following grammar
- $S \rightarrow aAd \mid bBd \mid aBe \mid bAe$
 $A \rightarrow C$
 $B \rightarrow C$
- (i) Construct LALR parsing table. (6)
- (ii) Is the grammar an LALR grammar? (2)
- (iii) Check whether the string 'bcd' parsed by LALR (1)Parser. (5)
13. (a) What is a syntax-Directed translation? Write the SDD for representing and generating three address statement for logical statements. Explain with an example.

Or

- (b) (i) What is a three-address code (TAC)? Explain its significance and types. Write the various ways of representing TAC. (7)
- (ii) Represent the following statement in various ways.
- $A = B + C * D - E/F.$ (6)

14. (a) With a neat algorithm, elaborate about the design of a code generator.

Or

(b) Elaborate about the various issues encountered during the design of a code generator.

15. (a) Consider the following code

```
count = 0;
S = sqrt (n);
for (i = 2; i <= s; i++)
    if (a [i]) {
        count ++;
        for (j = 2* i; j <= n; j = j + i)
            a [j] = FALSE;
    }
```

(i) Translate the program into three-address statements. (6)

(ii) Apply peephole optimization. (7)

Or

(b) Write three address code, Construct DAG and identify local common sub-expressions for the following code and after eliminating the common sub-expressions, re-write the basic block.

```
prod = 0;
i = 1;
do
{
    prod = prod + a [i] * b [i];
    i = i + 1;
} while (i <= 10);
```

PART C — (1 × 15 = 15 marks)

16. (a) With a neat sketch, explain how the following statement is processed by compiler. Write the output of each phase.

$SUM = A + B * (1+C/D) * 100$

Where

A,B,C are integers and D is a floating point number.

Or

- (b) Construct Operator Precedence parsing Table for the following grammar.

$E \rightarrow T$

$T \rightarrow TT+ | TT* | TT- | TT/ | (F)$

$F \rightarrow (E) | i.$