Registration No : | | | | | | | | | | |

Total Number of Pages : 04 **BTech/IDD(B.Tech and M.Tech**

RCS6C002

6th Semester Back Examination: 2024-25

COMPILER DESIGN

CSE, CSEAIML, CSEDS, CST

Time : 3 Hour

Max Marks : 100

Q.Code : S072

Answer Question No.1 (Part-1) which is compulsory, any eight from Part-II and any two from Part-III.

The figures in the right hand margin indicate marks.

Part-I

Q1 **Answer the following questions :** **(2 x 10)**

a) Define the role of a parser in syntax analysis phase.

b) Find the regular expression for the language L = {All strings of 0's and 1's whose first symbol is 0 and last symbol is 1 and length of the string is at least 4}.

c) State the work of linker and loader in language processing system.

d) Generate the string **w = abaaba** from the given grammar by applying leftmost derivation.

$A \rightarrow aBA \mid \epsilon$

$B \rightarrow AbB \mid AA \mid ba$

e) State the problems will arise if a grammar used for parsing is left recursive. Eliminate left recursion from the grammar:

$S \rightarrow S * S \mid T$

$T \rightarrow (S) \mid id$

f) Justify why the below grammar is not LL(1) grammar?

$S \rightarrow abSa \mid aaAb$

$A \rightarrow baAb \mid bb$

g) Distinguish between SDD and SDT.

h) List out possible lexical and syntax errors while compiling a program

i) Demonstrate the properties of code generation phase.

j) Suggest some examples of compiler.

Part-II

Q2 **Only Focused-Short Answer Type Questions- (Answer Any Eight out of Twelve)** **(6 × 8)**

a) Give the LL(1) parsing table for the below mentioned grammar find the FIRST & FOLLOW

E -> TE'
E' -> +TE' | ε
T -> FT'
T' -> *FT' | ε
F -> id | (E)

b) What is Peephole optimization? Explain with suitable example.

c) Consider the following grammar and check if the grammar is suitable for SLR parser. **S → AS | b**
A → SA | a

d) Consider the following grammar, Remove common prefix if any or do left factoring in the following grammars.

1. A → aAB | aBc | aAc
S -> bSSaaS |bSSaSb | bSb | a

e) Consider the following grammar and their Syntax Directed Translation rules:

CFG	Semantic Action
S → S * A	{ S.val = S.val * A.val}
S → A	{ S.val = A.val}
A → A + B	{ A.val = A.val – B.val}
A → B	{ A.val = B.val}
B → id	{ B.val = id.val}

Draw the annotated parse tree of the following SDT for the input string : **4 * 6 + 3 * 7** and evaluate it.

f) What is Bottom up Parsing? Show the handle of each right sentential form. S → C + E + C C → d/ad/b C → id

g) Discuss the rules to create Directed Acyclic Graph (DAG). Consider the following block of TAC, construct a DAG for it and find out the common expressions eliminated by DAG.

(1) a = b * c
(2) d = b
(3) e = d * c
(4) b = e
(5) f = b * c
(6) g = f + d

h) Illustrate canonical collection of LR(1) items for the following grammar.

S→AA
A→aA | b

i) Discuss different loop optimization techniques with suitable examples of each type.

j) Construct the nondeterministic finite automata for the regular expression: **q (p | q)* p**. Convert this NFA to DFA.

k) Consider the following block of TAC, construct a DAG for it and find out the common expressions eliminated by DAG.

(1) a = b * c

(2) d = b

(3) e = d * c

(4) b = e

(5) f = b * c

(6) g = f + d

l) List out the benefits of using intermediate code for the source code. For the expression **(a+b) * (c+d) –(a+b+c)** find its representation in quadruple, triple form

Part-III

Only Long Answer Type Questions (Answer Any Two out of Four)

Q3 **a)** Define cross compiler. Explain various phases of compiler design. **(8)**

b) Describe the output for the expression L → 0 + V – E ^ 2 after each step. **(8)**

Q4 **a)** Construct predictive parsing table for the following grammar and check if the grammar is LL(1) or not. **(8)**

E → TE'
E' →+TE' | ϵ
T → FT'
T' → *FT' | ϵ
F → (E) | id

b) Consider the following grammar and show the handle of each right sentential form for the string (a,(a,a)). For the input string (a,(a,a)), show various steps used in the parsing process by using a stack in shift reduce parser. **(8)**

S → (A) | a
A → A , S | S

Q5 **a)** Discuss syntax directed translation (SDT). For the following SDT, what is the output generated if SDT is carried out with the given input **a + b * c**. **(8)**

Rule	Semantic Action
E → E + T	{ printf(" + "); }
E → T	{ }
T → T * F	{ printf(" * "); }
T → F	{ }
F→ id	{ printf(id.val); }.

b) Write down the roles of a symbol table and error handler in compilation process. Briefly explain the lexical, syntactic and semantic errors that are handled by error handler **(8)**

Q6 **a)** What is the necessity of optimization in compilation? Discuss the different code optimization techniques **(8)**

b) Discuss the techniques used for error handling in YAAC compiler. Or, Write a sample lex pseudo code to recognize identifier, number, relational operators and whitespaces. **(8)**