13

S

140\c

CS/IS7G1

Reg. No. □□□□□□□□□□

Seventh Semester B.E. Degree Examination, February 2002

Computer Science/Information Science and Engineering

Compiler Design

me: 3 hrs.] [Max.Marks : 100

Note: *Answer any FIVE full questions.*

1. (a) Briefly explain different phases of a compiler and its function with a neat diagram. (12 Marks)

   (b) Explain the implementation of lexical analyzer with an example. (8 Marks)

2. (a) Construct a DFA using subset algorithm for regular expression $(a/b)^* abb$. (12 Marks)

   (b) Write the transition diagram recognising 'C' language reserved words

   i) Case ii) If iii) Struct iv) Switch

   v) Else if vi) Char vii) For viii) While. (8 Marks)

3. (a) Explain with an example the stack implementation of shift reduce parser. (10 Marks)

   (b) Determine the operator precedence relationship table for the grammer,
   $E \rightarrow E + E \,/\, E - E \,/\, E * E \,/\, E \uparrow E \,/\, (E) \,/\, -E \,/id$
   assuming

   i) ↑ is of higher precedence and right associative

   ii) * and / are next higher precedence and left associative

   iii) + and − are next lowest precedence and left associative. (10 Marks)

4. (a) Eliminate left recursion and calculate first and follow symbols for each non-terminal in the grammer,

   $E \rightarrow E * T/T$
   $T \rightarrow T/f/F$
   $F \rightarrow (e)/id$ (12 Marks)

   (b) Write an algorithm to eliminate left recursion for a grammer with no-cycle or ϵ - productions. (8 Marks)

5. (a) Briefly explain the working of a predictive parser. (8 Marks)

   (b) Construct the LALR parser table for the grammer given below

   $S' \rightarrow S$
   $S \rightarrow CC$
   $C \rightarrow cC \;/d$ (12 Marks)

6. (a) Explain different storage allocation strategies. (10 Marks)

   (b) What are the issues in the design of a code generator? Discuss in detail. (10 Marks)

7. (a) Consider the program segment given below

Contd.... 2

```
Test = 0
i = 1
do
    Test = Test + A[i] * B[i]
    i = i + 1
while (i ≤ 10)
```

i) Produce a 3-address code

ii) Obtain the basic blocks

iii) Obtain the flow graph. (12 Marks)

(b) Explain code optimization for the example in Q.NO.7(a). (8 Marks)

8. Write short notes on:

i) Context free grammer.

ii) Recursive decent parsing.

iii) The DAG representation.

iv) LR parser. (4×5=20 Marks)

** * **

4 1H0/c

CS/IS7G1

(11)

Reg. No. □□□□□□□□□□

Seventh Semester B.E. Degree Examination, July/August 2002

Computer Science/Information Science and Engineering

Compiler Design

Time: 3 hrs.] [Max.Marks: 100

Note: 1. *Answer any FIVE full questions.*
2. *All questions carry equal marks.*

1. (a) Distinguish between compilers and interpretets and compare their relative merits and demerits. (8 Marks)

   (b) Explain the need for multiple passes in compilers. (6 Marks)

   (c) Give a technique for reducing number of passes. (6 Marks)

2. (a) Briefly explain the role of a transition diagram in designing a lexical analyzer. (7 Marks)

   (b) Construct a transition diagram to recognize the following set of tokens.
   BEGIN END DO identifier.
   Give the program segments for
   i) Start state and ii) An accepting state. (7 Marks)

   (c) Give LEX specification to recognize integer constants. (6 Marks)

3. (a) Give an unambiguous grammar for if-then-else construct in which 'else' is optional. (5 Marks)

   (b) Briefly explain two programming constructs which cannot be modelled by context free grammars. (6 Marks)

   (c) Distinguish between top down and bottom up parsing. (2 Marks)

   (d) Briefly explain the problems that arise in top down parsing. (7 Marks)

4. (a) Construct SLR(1) parsing table for the grammar below.

   $E \longrightarrow T * E | T$

   $T \longrightarrow T + id | id$ (10 Marks)

   (b) Show that when 'similar' states of LR(1) parser are merged to obtain LALR(1) tables no new shift/reduce conflicts are introduced whereas reduce/reduce conflicts may arise. (5 Marks)

   (c) Compare relative merits and demerits of LL(1) and LR(1) parsers. (5 Marks)

5. (a) Briefly explain the concept of syntax directed translation scheme. (6 Marks)

   (b) Add semantic actions to process c-declarations given the grammar below.

   $D \longrightarrow TL;$

   $T \longrightarrow int | float;$

   $L \longrightarrow L, id | id$ (6 Marks)

Contd.... 2

(c) Briefly explain the use of attribute stack during bottom-up evaluation of synthesized attributes.
(8 Marks)

6. (a) Convert the following segment into 3-address code and identify the basic blocks

while $(A \leq C)$ do
if $(A = C)$ then $x_i = x + 1$ else $y_1 = y + 1$
(10 Marks)

(b) Briefly explain any four kinds of machine independent code optimization.
(10 Marks)

7. (a) Briefly explain the issues that arise in code generation.
(10 Marks)

(b) Give a suitable storage allocation scheme for c-like language along with the structure of activation record.
(10 Marks)

8. Write short notes on:

i) YACC

ii) Display

iii) Peep-hole optimization

iv) Precedence functions.

(4×5=20 Marks)

** * **

1140|C

3

CS/IS7G1

10

Reg. No. □□□□□□□□□□

Seventh Semester B.E. Degree Examination, January/February 2003

Computer Science/Information Science and Engineering

Compiler Design

Time: 3 hrs.] [Max.Marks : 100

Note: *1. Answer any FIVE full questions.*
2. All questions carry equal marks.

1. (a) Describe an interpreter. Discuss its relative merits and demerits over compiler. (10 Marks)

(b) Construct a transition diagram to accept the set of tokens given below:

BESIN

END

ELSE

identifier

Give the program segments for

i) Start State ii) any one final state (10 Marks)

2. (a) Give a grammar for expressions using the operators + and * such that

i) * has less priority than +

ii) * is right associative

iii) + is left associative.

iv) Parenthesized expression has highest priority. (5 Marks)

(b) Show that the grammar

$S \longrightarrow ictses|icts|a$

is ambiguous.

Give an unambiguous grammar for the same language. (9 Marks)

(c) Given the grammar

$S \longrightarrow a\ |(L)$

$L \longrightarrow L, S|S$

give i) Left most derivation, and

ii) Right most derivation for the string

$\Big((a,a),a\Big)$ (6 Marks)

3. (a) Given the grammar

$$E \longrightarrow T + E|T$$
$$T \longrightarrow T * F|F$$
$$F \longrightarrow id|(E)$$

i) Remove left recursion and do left factoring if needed.

ii) For the resulting grammar construct LL(1) parsing table. (12 Marks)

(b) Briefly explain, with an example, how operation precedence relations can be used to identifying the handle. (8 Marks)

4. (a) Construct canonical LR(1) parsing table for the grammar

$$E \longrightarrow E * T|T$$
$$T \longrightarrow T + id|id$$ (10 Marks)

(b) Compare the relative merits and demerits of SLRCD, LALRCD, and canonical LRCD parsers. (10 Marks)

5. (a) Briefly describe the concept of syntax directed definition. (8 Marks)

(b) Briefly explain the method of evaluation of S-attributed definition during bottom-up parsing. Illustrate it on the definition below.

$$E \rightarrow E^{(1)} * E^{(2)}\{E.val = E^{(1)}.val * E^{(2)}.val\}$$
$$E \rightarrow E^{(1)} + E^{(2)}\{E.val = E^{(1)}.val * E^{(2)}.val\}$$
$$E \rightarrow Const\{E.val = Const.lval\}$$

Input : $4 * 5 + 6$ (12 Marks)

6. (a) Give SDTS for translation of boolean expression to 3-address code which uses jump statements to avoid redundant evaluation. (8 Marks)

(b) Give a scheme for runtime storage allocation for C-like languages along with the structure of activation record. Briefly describe the actions required during procedure call and return. (12 Marks)

7. (a) Briefly explain any 5 kinds of code optimisation with an example each. (10 Marks)

(b) Generate code for PDP-11 like machine for the statement.

$$A \longleftarrow B * C + D * E * C$$

[Use GETREG( ) function. Assume that only 2 registers R_0 and R_1 are available.] (10 Marks)

8. Write short notes on

(a) Display (b) Peep hole optimization
(c) LEX (d) Error recovery in LR parsers.

(5×4=20 Marks)

** * **

2 10

140/e

26

CS/IS7G1

USN □□□□□□□□□□

Seventh Semester B.E. Degree Examination, June 2003

Computer Science/Information Science and Engineering

Compiler Design

Time: 3 hrs.] [Max.Marks : 100

Note: *1. Answer any FIVE full questions.*
2. All questions carry equal marks.

1. (a) Explain with the block diagram, different phases of a compiler. (10 Marks)

   (b) Distinguish between a phase and a pass. (2 Marks)

   c) Briefly explain the need of multipass in a compiler. (8 Marks)

2. (a) Briefly explain the use of transition diagram in constructing a lexical analyzer with an example. (10 Marks)

   (b) Show that the grammar below is ambiguous.

   $$E \rightarrow E * E | E + E | (E) | id$$

   Give an unambiguous grammar for the above language such that.

   i) + has higher priority.

   ii) * and + are right associative. (10 Marks)

3. (a) Briefly explain the problems associated with top-down parsing. (6 Marks)

   (b) Define LL(I) grammar. Give conditions for a grammar to be LL(I) based on FIRST and FOLLOW sets (6 Marks)

   (c) Construct LL(I) parsing table for the grammar.

   $S \rightarrow ictSS^1/a$

   $S^1 \rightarrow eS/\in$ (8 Marks)

   Is the grammar LL(I)?

4. (a) Construct SLR(I) parsing table for the grammar.

   $E \rightarrow E \not+ T/T$

   $T \rightarrow id + T/id$ (8 Marks)

   (b) Give the limitations of SLR(I) praser. (4 Marks)

   (c) Briefly explain the error detection and recovery in LR parsers. (8 Marks)

5. (a) Briefly explain the concept of syntax directed definition with an example. (10 Marks)

   (b) For the SDD given below, construct annotated parse tree for the input

   $Real\ \ a, b, c;$

   $D \rightarrow TL; \{L.in = T.s\}$

   $L \rightarrow L_1, id\{L_1.in = L.in; id.type = L.in;\}$

   $L \rightarrow id\{id.type = L.in\}$ (10 Marks)

Contd.... 2

6. (a) Give SDTS for translation of arithmetic expressions into quadruples. Give the output for $a * b + C * d$ (10 Marks)

   (b) Briefly explain the concept of 'static scope' and the use of display in accessing global variables. (10 Marks)

7. (a) Briefly explain any 5 kinds of code optimization. (10 Marks)

   (b) Briefly explain a 'simple code generator' which makes use of 'next use' information in the use of available registers. (10 Marks)

8. Write short notes on :

   a) LEX

   b) DAG representation of basic blocks.

   c) Recursive descent parsing

   d) Precedence functions.

   (4×5=20 Marks)

** * **

USN □□□□□□□□□□

Seventh Semester B.E. Degree Examination, January/February 2004

Computer Science/Information Science and Engineering

Compiler Design

Time: 3 hrs.] [Max.Marks : 100

Note: 1. *Answer any FIVE full questions.*
2. *All questions carry equal marks.*

1. (a) Briefly explain the need for multi-pass in a compiler. (6 Marks)

(b) Briefly explain a strategy to reduce no of passes. (4 Marks)

(c) Write a transition diagram to recognize the following set of tokens. Write program segments for start state, any one intermediate state and any one final state.

BEGIN
END
ELSE
Identifier (10 Marks)

2. (a) Show that the following grammar to specify if then-else statement is ambiguous.

$S \rightarrow ictS|ictS \mid Q$

Give an equivalent unambiguous grammar. (6 Marks)

(b) Define FIRST and FOLLOW. Explain how they are used in checking LL(1) conditions. (6 Marks)

(c) Given the grammar

$E \rightarrow E + T \mid T$
$T \rightarrow id + T \mid id$

Construct LL(1) parsing table after making necessary modifications. (8 Marks)

3. (a) Construct SLR(1) parsing table for the grammar given below

$E \rightarrow T * E/T$
$T \rightarrow T + F/F$
$F \rightarrow id$ (10 Marks)

(b) Compare the relative merits and demerits of LL(1), SLR(1), LALR(1), and canonical LR(1) parsing methods. (10 Marks)

4. (a) Briefly explain the concept of syntax directed definition with an example.(8 Marks)

(b) Briefly explain the use of attribute stack in evaluation of s-attributed SDD. Illustrate it using the example of syntax directed definition of a desk calculator and the input $3 + 4 * 5$ (12 Marks)

5. (a) Give the syntax directed translation scheme for translation of boolean expression and if-then-else statement (use the control flow 'method' for translation of Boolean expression. Illustrate the method on the input

if $a < b$ and $c < d$ then $a_i = b + c$ else $b_i = b + d$ (12 Marks)

(b) Briefly explain the run time storage allocation scheme for a C-like language. (8 Marks)

Contd.... 2

6. (a) Distinguish between "Static scope" and "Dynamic scope". Briefly explain the use of display in accessing global variables (in static scope). (10 Marks)

   (b) Briefly explain any 5 kinds of code optimization. (10 Marks)

7. (a) Briefly explain the main issues in code generation. (10 Marks)

   (b) Briefly explain a scheme for code generation for PDP-11 like instruction format. The scheme should use "next use" information in selecting a suitable register. Illustrate it on the statement

   $$a_i = a * b + c * d$$

   Assume that only 2 registers are available. (10 Marks)

8. Write short notes on :

   (a) LEX

   (b) Recursive descent parsing

   (c) Peep hole optimization

   (d) L-attributed SDD.

(5×4=20 Marks)

** * **

CS664

NEW SCHEME

USN □□□□□□□□□□

Sixth Semester B.E. Degree Examination, July/August 2005

Computer Science / Information Science and Engineering

Compiler Design

Time: 3 hrs.] [Max.Marks : 100

Note: *Answer any FIVE full questions.*
2. All questions carry equal marks.

1. (a) What is the need for a multipass compiler? Explain. (6 Marks)
   (b) Briefly explain a strategy to reduce the number of passes. (4 Marks)
   (c) Construct a transition diagram to recognize the tokens given below :
      i) Keywords : BEGIN END
      ii) Integer constant
      iii) Identifier
   Show the program segment for the start state and one final state. (10 Marks)

2. (a) Define :
      i) left most derivation
      ii) right most derivation and
      iii) Parsetree (6 Marks)
   (b) Your are given the grammar :
   $S \to a|b|(L)$
   $L \to L, S|S$
   Give left most derivation for the sentence ((a, b), a) (4 Marks)
   (c) Define ambiguity and show that the grammar below is ambiguous (6 Marks)
   $E \to E + E|E * E|(E)|id.$
   (d) Give an unambiguous grammar for the above language such that
      i) + has higher priority
      ii) $*$ has less priority
      iii) Both are right associative. (4 Marks)

3. (a) Briefly explain the problems associated with top-down parsing. (6 Marks)
   (b) Given the grammar
   $E \to T + E|T$
   $T \to T * F|F$
   $F \to (E)|id$
      i) Make the necessary changes to make it suitable for u(1) parsing.
      ii) For the resulting grammar, construct FIRST and FOLLOW sets and LL(1) parsing table. (10 Marks)
   (c) Briefly explain the rules for building operator precedence relations from priority and associativity. (4 Marks)

Contd.... 2

4. (a) Given the grammar

$$E \rightarrow E + T | T$$
$$T \rightarrow T * id | id$$

i) Construct sets of LR (1) items

ii) Construct canonical LR (1) parsing table

iii) Is the grammar LALR (1)? Justify your answer (8+4+2=14 Marks)

(b) Write a note on error recovery in LR parsers. (6 Marks)

5. (a) Briefly explain the concept of syntax directed definition. (4 Marks)

(b) Define synthesized and inherited attributes. (2 Marks)

(c) Define S-attributed and L-attributed SPD's (4 Marks)

(d) Give the syntax directed definition to process a simple variable declaration in C and construct annotated parse tree for the input

int a, b, c; (10 Marks)

6. (a) Briefly explain the run time storage scheme for C-language. Give the structure of the activation record. Briefly explain the actions required during.

i) function call

ii) function beginning

iii) return

iv) after the return (2+4+2×4=14 Marks)

(b) Briefly explain the difference between static scope and dynamic scope. (6 Marks)

7. (a) Briefly explain any five kinds of source level optimization. (10 Marks)

(b) Given the program segment :

```
a = b + c + d;
b = c+ a;
c = c+ d;
```

i) Give the 3 - address code

ii) Construct 'next use' information. Assume that temporaries are 'dead' and other variables are 'live' at the end. (10 Marks)

8. Write short notes on :

a) Panic modes

b) Peep hole optimization

c) Call-by-name

d) Register allocation by graph colouring (5×4=20 Marks)

** * **

NEW SCHEME

CS664

6

Reg. No. □□□□□□□□□□

Sixth Semester B.E. Degree Examination, January/February 2006

Computer Science and Engineering

Complier Design

Time: 3 hrs.) (Max.Marks : 100

Note: ***Answer any FIVE full questions.***

1. **(a)** Explain with diagram the phases of a compiler. **(8 Marks)**

(b) Show the translation made by each of these phases for the statement $a := b+c*10$ where a,b and c are reals. **(4 Marks)**

(c) Write a Lex program that recognises the following :

i) if, then, else, begin, end

ii) relational operators $<, >, <=, >=$

iii) numbers in integer and reals separately

iv) identifier (letter followed by digit)

Also find the number of occurences of the above mentioned in the input file. **(8 Marks)**

2. **(a)** Construct the operator precedence relation table for the grammar given that

$E \to E+E \mid E-E \mid E*E \mid E \uparrow E \mid (E) \mid id \mid EdivE$

i) ↑ has highest precedence and right associative

ii) * and div operators are of next higher precedence and are left associative

iii) + & − are lowest precedence and are left associative.

Also find the precedence functions f and g. **(10 Marks)**

(b) Explain left recursion and show how it is eliminated. Eliminate left recursion from the following grammar

$E \to E+T \mid T$
$T \to T*F \mid F$
$F \to (E) \mid id$

Also describe the algorithm used for eliminating left recursion. **(10 Marks)**

3. **(a)** What is left factoring? Show how we can left factor the grammar.

$S \to iEtSeS \mid iEtS \mid a$
$E \to b$ **(5 Marks)**

(b) Define LL(1) grammar. Give conditions for a grammar to be LL(1) based on first and follow. **(5 Marks)**

(c) Obtain SLR parsing table for the grammar given in question 2(b). **(10 Marks)**

Contd.... 2

4. (a) Given the grammar

$$E \rightarrow E + T \mid T$$
$$T \rightarrow id + T \mid id$$

Construct LL(1) parsing table after making necessary modification. Is the grammar LL(1)? **(8 Marks)**

(b) Obtain the syntax directed definition for a simple desk calculator. Obtain an annotated parse tree for the expression 4*8+6n. **(8 Marks)**

(c) Describe S attributed and L attributed definition. **(4 Marks)**

5. (a) What is an activation record? Explain the purpose of each item in the activation record with example. **(8 Marks)**

(b) Explain briefly different storage allocation strategies. **(8 Marks)**

(c) Explain with examples quadruples, triples and indirect triples. **(4 Marks)**

6. (a) Explain the issues involved in code generation. **(10 Marks)**

(b) Write the code generation algorithm and generate code for the following expression.

$W = (X - Y) + (X - Z) + (X - Z)$. **(10 Marks)**

7. (a) Explain any five kinds of code optimisation techniques. **(10 Marks)**

(b) What are basic blocks and how do you partition a 3 address code into the basic blocks? **(5 Marks)**

(c) Translate the following into 3 address code

While $(A < B)$ do

if $(C < D)$ then $X = Y + Z$ **(5 Marks)**

8. Write short notes on any FOUR :

(a) Handle pruning

(b) Yacc

(c) Global register allocation

(d) DAG's

(e) Next use information for register allocation. **(5×4=20 Marks)**

** * **

15·

22

CS664 (5)

USN □□□□□□□□□□

NEW SCHEME

Sixth Semester B.E. Degree Examination, July 2006
Computer Science and Engineering
Compiler Design

Time: 3 hrs.] [Max. Marks:100

Note: *1. Answer any FIVE full questions.*

1 a. With the help of a neat diagram explain the various phases of a compiler and explain its functioning. **(10 Marks)**

b. Why are look ahead operators required? With reference to this, give Lex Specification to recognize the DO and IF statement keywords in Fortran 77. **(10 Marks)**

2 a. Consider the grammar,

$R \to R+R|R.R|R*|(R)|a|b|c$

i) Show that the grammar is ambiguous.

ii) Convert the above to equivalent unambiguous grammar.

iii) Write the parse structure in both grammar for the string a+b*.c **(10 Marks)**

c. Give an algorithm for constructing predictive parsing table. Apply this algorithm for the following grammar to obtain parsing table.

$E \to TE^{1},\ \ T \to FT^{1},\ \ T^{1} \to *FT^{1} |\in,\ \ F \to (E)\,|\,id$. **(10 Marks)**

3 a. Write an operator precedence parsing algorithm. **(08 Marks)**

b. Obtain set of canonical LR(0) items for the grammar $E^{1} \to E,\ \ E \to E+T\,|\,T,$ $T \to T*F\,|\,F,\ \ F \to (E)\,|\,id$ **(08 Marks)**

c. Write a note on Yale – Parser generator. **(04 Marks)**

4 a. Obtain directed acyclic graph for the expression $a+a*(b-c)+(b-c)*d$. Also give sequence of instructions for constructing above dag. **(06 Marks)**

b. Give Syntax Directed Definition with inherited attribute L.in. Show annotated parse tree for the sentence real id_1, id_2, id_3. **(08 Marks)**

c. Construct syntax tree for expression $a-4*C$ and give the sequence of functions calls. **(06 Marks)**

5 a. Write a note on activation tree in source language issues. **(04 Marks)**

b. Explain Stack and Heap allocation strategies with the help of necessary diagrams. Also highlight their differences. **(06 Marks)**

c. Explain different dynamic storage allocation techniques. **(10 Marks)**

6 a. Explain implementation of 3 – address code with an example. **(06 Marks)**

b. Give annotated parse tree for the string x_i $A[y,z]$ **(06 Marks)**

c. Explain methods of transformation of Boolean expressions. Generate 3 address code for the Boolean expression a<b or c<d and e<f. **(08 Marks)**

7 a. ˙Explain the different issues involved in the design of code – generator. **(10 Marks)**

b. Explain different types of transformation on basic blocks. **(06 Marks)**

c. Obtain code sequence generated for pointer assignment $a := *P$ and $*P := a$. **(04 Marks)**

8 a. Explain graph colouring technique for allocation register. **(04 Marks)**

b. Explain any five kinds of code optimization techniques. **(10 Marks)**

c. Explain how the following code can be optimized and give its dag.

a: = b + c
b: = a – d
c: = b + c
d: = a – d. **(06 Marks)**

*****

CS664

USN □□□□□□□□□□

NEW SCHEME

Sixth Semester B.E. Degree Examination, Dec. 06 / Jan. 07

CS / IS

Compiler Design

Time: 3 hrs.] [Max. Marks:100

Note : ***Answer any FIVE full questions.***

1 a. What is compiler? Explain the different phases of compiler by considering the following statement as input.

position := initial + rate * 60 **(10 Marks)**

b. Briefly explain the need for multipass in compiler. **(06 Marks)**

c. Briefly explain a strategy to reduce the number of passes. **(04 Marks)**

2 a. Write a transition diagram to recognize the following set of tokens. Write program segments for start state, any one of the intermediate states and any one final state.

BEGIN
END
ELSE
Identifier **(10 Marks)**

b. What is look ahead operator? With examples show how this operator may be used to solve lexical analysis problems. **(10 Marks)**

3 a. With a schematic, explain the role of Parser. List and explain various error recovery strategies. **(10 Marks)**

b. Define left-recursion. Eliminate left recursion from the following grammar :

$E \rightarrow E + T / T$
$T \rightarrow T * F / F$
$F \rightarrow (E) / id$

Also obtain FIRST and FOLLOW symbols for the above resulting grammar. **(10 Marks)**

4 a. Construct SLR(1) parsing table for the following grammar

$E \rightarrow T * E / T$
$T \rightarrow T + F / F$
$F \rightarrow id$ **(10 Marks)**

b. Compare the relative merits and demerits of LL (1), SLR (1), LALR (1) and canonical LR (1) parsing methods. **(10 Marks)**

5 a. Briefly explain the concept of syntax directed definition with example. **(06 Marks)**

b. Write a note on L-attributed definition. **(04 Marks)**

c. Give SDTS for an arithmetic expression with +, * and -. Show annotated parse tree for the input 3 + 4 * 5. **(10 Marks)**

Contd.... 2

6 a. Explain in detail, different storage allocation strategies. **(10 Marks)**

b. With example explain different parameter passing methods. **(10 Marks)**

7 a. Briefly explain the main issues in code generation. **(10 Marks)**

b. Briefly explain any five kinds of code-optimization. **(10 Marks)**

8 Write short notes on :

a. LEX

b. Recursive descent parser.

c. Dead code elimination.

d. L-attributed SDD. **(20 Marks)**

*****

CS664

USN □□□□□□□□□□

NEW SCHEME

Sixth Semester B.E. Degree Examination, July 2007

CS / IS

Compiler Design

e: 3 hrs.] [Max. Marks:100

Note : *Answer any FIVE full questions.*

a. With a neat diagram, explain the different phases of compilation. **(10 Marks)**

b. Discuss the different compiler construction tools. **(05 Marks)**

c. Explain the input buffering strategy used in lexical analysis phase. **(05 Marks)**

a. Define the following terms:

i) Token
ii) Lexeme
iii) Pattern
iv) Handle. **(04 Marks)**

b. Write the transition diagram that accepts the reserve words case, const, char, and continue. **(04 Marks)**

c. Write the principle of non recursive predictive passing. **(07 Marks)**

d. Show that the following grammar G, is ambiguous. $A \rightarrow AA \mid (A) \mid \in$. **(05 Marks)**

a. What is left recursion and left factoring, explain with an example for each? **(08 Marks)**

b. Compute first and follow for the following grammar after eliminating left recursion.
$E \rightarrow E + T / T \quad T \rightarrow TF / F \quad F \rightarrow F * \mid a \mid b \mid (\in)$ **(08 Marks)**

c. Explain the different error handling strategies of the parser. **(04 Marks)**

a. Generate SLR parsing table for the following grammar $S \rightarrow (L) / a \quad L \rightarrow L, S/S$ **(08 Marks)**

b. Write the precedence function for the grammar a. $S \rightarrow (L) / a \quad L \rightarrow L, S / S$ **(07 Marks)**

	a	(	)	,	$
a			$\gtrdot$	$\gtrdot$	$\gtrdot$
(	$\lessdot$	$\lessdot$	$\doteq$	$\lessdot$	
)			$\gtrdot$	$\gtrdot$	$\gtrdot$
,	$\lessdot$	$\lessdot$	$\gtrdot$	$\gtrdot$	
$	$\lessdot$	$\lessdot$			

c. Write the algorithm for construction of LR parsing table. **(05 Marks)**

a. Construct LR(1) items for the grammar G, $A \rightarrow (A) / a$ **(08 Marks)**

b. Define synthesized attribute and inherited attribute. Give the semantic rule for implementation of desk calculator with example. **(07 Marks)**

c. Construct annotated parse tree for the expression $4 * (3 + 5) - 7$ using semantic rule for simple desk calculator. **(05 Marks)**

Contd....2

6 a. Write a note on activation records and its use. **(05 Marks)**

b. Explain in detail the stack allocation strategy. **(08 Marks)**

c. What is printed by the program assuming

i) Call-by-value

ii) Call-by-reference

iii) Copy-restore linkage

iv) Call-by-name

```
Program main (input, output);
Procedure P(x, y, z)
begin
        y : = y + 1;
        z : = z + x ;
end
begin
        a : = 2;
        b : = 3;
        P(a+b, a, a);
        Print a
end.
```

(07 Marks)

7 a. Write the semantic action for

i) do while loop

ii) for loop **(08 Marks)**

b. Define the following terms :

i) Usage count.

ii) Register assignment and allocation.

iii) Register descriptor and address descriptor. **(06 Marks)**

c. Explain different loop optimization techniques. **(06 Marks)**

8 a. Generate the machine code and compute the cost for the following instruction set. Assume there are only two registers available for storing intermediate value

a = 1 b = 10 c = 20

d = a + b; e = c + d; f = c + a; e = b + d; d = 5 + f **(10 Marks)**

b. Construct the basic block and flow graph for the following program statement

```
begin
        locn = -1
        i = 0
        while (i < 100) do
        begin
                if a(i) = x then locn = I
                i = i + 1
        end
end
```

(10 Marks)

*****

7

76

CS664

USN □□□□□□□□□□

Sixth Semester B.E Degree Examination, Dec. 07 / Jan. 08

Compiler Design

Time: 3 hrs. Max. Marks:100

Note : *Answer any FIVE full questions.*

1 a. Explain the different phases of a compiler with a block diagram. (10 Marks)

b. Construct transition diagram for the following: i) Relational operators ii) Identifiers and keywords iii) Unsigned numbers. (06 Marks)

c. Construct a NFA for regular expression $(a / b)^*$ abb (04 Marks)

2 a. Define Ambiguity. Show that the following grammar is ambiguous : (08 Marks)

$E \rightarrow E + E / E - E / E * E / (E) / id$

b. Given the grammar: $E \rightarrow E + T / T$; $T \rightarrow T * F / F$; $F \rightarrow (E) / id.$

i) Remove left recursion. (12 Marks)

ii) For the resulting grammar, construct LL (1) parsing table.

3 a. Determine the operating precedence relation table for the grammar :

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid E \uparrow E \mid (E) \mid - E \mid id$, assuming

i) $\uparrow$ is of highest precedence and right –associative.

ii) * and / are of next highest precedence and left associative and (10 Marks)

iii) + and – are of lowest precedence and left - associative.

b. Construct Canonical LR (1) parsing table for the grammar. (10 Marks)

$E \rightarrow E + T / T$; $T \rightarrow T * F / F$; $F \rightarrow (E) / id$

4 a. Construct LALR parsing table for the grammar : $S \rightarrow C C$; $C \rightarrow c C / d.$ (10 Marks)

b. Briefly explain the concept of syntax directed definition with an example. (10 Marks)

5 a. Explain L – attributed definition in detail. (10 Marks)

b. Briefly explain the different data structures used for symbol table. (10 Marks)

6 a. Briefly explain the different types of intermediate codes with an example. (10 Marks)

b. Explain the structure preserving transformation on Basic blocks. (10 Marks)

7 a. Explain in detail various issues involved in Code Generation phase. (10 Marks)

b. Briefly explain any five kinds of code optimization with an example each. (10 Marks)

8 Write short notes on:

a. LEX

b. Recursive Descent Parsing.

c. Error recovery in Operator – precedence parsing. (20 Marks)

d. DAG representation of Basic blocks.

*****

6·

13

USN | | | | | | | | | |

06IS662

Sixth Semester B.E. Degree Examination, June-July 2009

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. What are the various phases of compiles? Explain each phase in detail. Write down the output of each phase for the expression "position = initial + rate*60". **(14 Marks)**

b. Define Token, Pattern and lexeme? Explain the role of the lexical analyzer with block diagram. **(06 Marks)**

2 a. What are the Error – Recovery strategies in parsing? Explain briefly. **(04 Marks)**

b. Prove that the following grammar is ambiguous for the string "if E_1 then if E_2 then S_1 else S_2".

$$\begin{aligned} \text{stmt} \rightarrow\ & \text{if expr then stmt} \\ |\ & \text{if expr then stmt else stmt} \\ |\ & \text{other} \end{aligned}$$

(04 Marks)

c. Define context free grammar and remove the left recursion from the grammar.

$$S \rightarrow Aa \mid b$$
$$A \rightarrow Ac \mid Sd \mid \varepsilon$$

(04 Marks)

d. Compute FIRST and FOLLOW for the grammar :

$$E \rightarrow TE^1$$
$$E^1 \rightarrow +TE^1 \mid \varepsilon$$
$$T \rightarrow FT^1$$
$$T^1 \rightarrow *FT^1 \mid \varepsilon$$
$$F \rightarrow (E) \mid id$$

(08 Marks)

3 Consider the following grammar :

$$E \rightarrow E + T$$
$$E \rightarrow T$$
$$T \rightarrow T * F$$
$$T \rightarrow F$$
$$F \rightarrow (E)$$
$$F \rightarrow id$$

a. Construct LR(o) automation for the above grammar. **(12 Marks)**

b. Show the moves of the parser for the input string id*id **(08 Marks)**

4 a. Write an algorithm to contract canonical – LR parsing tables. **(08 Marks)**

b.

$$S^1 \rightarrow S$$
$$S \rightarrow CC$$
$$C \rightarrow {}_cC \mid d$$

For the given grammar find out the LR(1) items using LALR parsing method. **(12 Marks)**

PART – B

5 a. i) What is Syntax-Directed Definition? Write the Semantic-Rules for the grammar

1) $L \rightarrow En$
2) $E \rightarrow E_1 + T$
3) $E \rightarrow T$
4) $T \rightarrow T_1 * F$
5) $T \rightarrow F$
6) $F \rightarrow (E)$
7) $F \rightarrow digit$

ii) Option Annotated parse tree for 3*5+4n **(10 Marks)**

b. Explain the application of syntax. Directed Translation with an example. **(10 Marks)**

6 a. Obtain a directed Acyclic Graph and three address code for the expression

$a + a * (b - c) + (b - c) *d$ **(06 Marks)**

b. Obtain the Quadraples and triples for above expression. **(04 Marks)**

c. Explain briefly the translation of expression with an example. **(10 Marks)**

7 Write short notes on :

a. Storage Organization

b. Activation Tree and Activation Record

c. Memory Manager

d. Performance Metrics in Garbage Collection **(20 Marks)**

8 a. Obtain the optimal machine code for the following three-address code sequences.

t = a + b	t = a + b
t = t * c	t = t + c
t = t / d	t = t / d

(06 Marks)

b. What are Basic blocks and flow graphs? Write an algorithm to partionting three-address instructions into basic blocks. **(06 Marks)**

c. Mention the different types of optimization of basic blocks? Explain any two with example. **(08 Marks)**

* * * * *

1

USN 1 3 P a p e r S h e e t s 5

06CS63

Sixth Semester B.E. Degree Examination, June-July 2009

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. What is meant by input buffering? Explain the use of sentinels in recognizing tokens. **(04 Marks)**

b. With the help of a diagram, explain the various phases of a compiler. **(08 Marks)**

c. Construct the transition diagram to recognize the tokens given below:
i) Identifier ii) Relational operator iii) Unsigned number. **(08 Marks)**

2 a. Briefly explain the problems associated with top-down parser. **(04 Marks)**

b. Show that the grammar below is ambiguous.

$E \rightarrow E + E \mid E * E \mid id$

Give an unambiguous grammar for the above grammar such that + has higher priority, * has less priority and both are right associative. **(08 Marks)**

c. Given the grammar

$S \rightarrow a \mid (L)$
$L \rightarrow L, S \mid S$

i) Do the necessary changes to make it suitable for LL(1) parser.
ii) Check the resultant grammar is LL(1) or not. **(08 Marks)**

3 a. What is a shift reduce parser? Explain the conflicts that may occur during shift reduce parsing. **(04 Marks)**

b. Given the grammar

$A \rightarrow (A) \mid a$

i) Find LR(0) items ii) Construct SLR(1) parsing table. **(08 Marks)**

c. Write SLR(1) parsing algorithm. Using the table constructed in Q.3(b), show the parsing steps for the string ((a)). **(08 Marks)**

4 a. Write YACC specification to perform arithmetic operation. **(04 Marks)**

b. Write an algorithm for constructing the canonical LR(1) parsing table. Construct canonical LR(1) parsing table for $S \rightarrow CC$
$C \rightarrow eC \mid d$ **(16 Marks)**

PART – B

5 a. Give the syntax directed definition to process a sample variable declaration in C and construct dependency graph for the input float x, y, z. **(10 Marks)**

b. Assuming suitable syntax directed definition, construct a syntax tree for the expression $a - 4 + e$. **(10 Marks)**

6 a. Describe the method of generating intermediate code for the flow control statements. **(10 Marks)**

b. Explain the different types of representation of 3-address code. Generate 3-address code for $a < b$ or $c > d$ or $e < f$. **(10 Marks)**

7 a. Explain in detail different dynamic storage allocation strategies. **(10 Marks)**

b. Distinguish between static scope and dynamic scope. Briefly explain access to non-local names in static scope. **(10 Marks)**

8 a. Explain the code generation algorithm and generate code for the following expression

$X = (a-b)+(a+c)$ **(10 Marks)**

b. Briefly explain main issues in code generation. **(10 Marks)**

4·

USN □□□□□□□□□□

06CS63

Sixth Semester B.E. Degree Examination, Dec.09/Jan.10

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting at least TWO questions from each part.***

Important Note: 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and/or equations written eg, 42+8=50, will be treated as malpractice.

PART – A

1 a. Explain a language processing system, with a block diagram. **(08 Marks)**

b. Explain the concept of input buffering in the lexical analysis. **(06 Marks)**

c. Write the transition diagram to recognize the token relop. (Corresponding to relational operators in a language). **(06 Marks)**

2 a. What is left-recursion? Eliminate left recursion from the following grammar:
$E \rightarrow E+T/T$; $T \rightarrow T*F/F$; $F \rightarrow (E)/id$ **(06 Marks)**

b. Obtain the predictive parsing table for the following grammar:
$S \rightarrow iEtSS'/a$; $S' \rightarrow eS/\in$; $E \rightarrow b$ **(14 Marks)**

3 a. Obtain LR(O) items for the following grammar:
$S \rightarrow L=R/R$; $L \rightarrow *R/id$; $R \rightarrow L$ **(08 Marks)**

b. Obtain first and follow symbols for the grammar shown in Q3 (a) and obtain SLR parsing table. Is the grammar SLR? **(12 Marks)**

4 a. Given the following grammar:
$S \rightarrow CC$; $C \rightarrow cC/d$
i) Construct sets of LR(1) items.
ii) Construct canonical LR(1) parsing table. **(12 Marks)**

b. Construct LALR parsing tables for the grammar shown in Q4 (a) using LR(1) items. **(08 Marks)**

PART – B

5 a. Explain the concept of syntax directed translation, with examples. **(06 Marks)**

b. Define inherited and synthesized attributes. **(04 Marks)**

c. Give SDD of a simple desk calculator. **(04 Marks)**

d. Write the annotated parse tree for $3*5+4n$. **(06 Marks)**

6 a. Draw the DAG for the arithmetic expression, $a+a*(b-c)+(b-c)*d$. Show the steps for constructing the DAG. **(10 Marks)**

b. What are three address codes? Explain different ways of representing three address codes, with examples. **(10 Marks)**

7 a. What is an activation record? Explain the purpose of each item in the activation record, with example. **(08 Marks)**

b. Distinguish between static scope and dynamic scope. **(04 Marks)**

c. What do you mean by calling sequence? Explain the actions performed during, i) function call ii) return. **(08 Marks)**

8 a. Explain the main issues in code generation. **(10 Marks)**

b. For the following program segment:

```
for i = 1 to 10 do
    for j = 1 to 10 do
        a[i, j] = 0.0
for i = 1 to 10 do
    a[i, i] = 1.0
```

generate intermediate code and identify basic blocks. **(10 Marks)**

* * * * *

3

3

14

USN □□□□□□□□□□ **06IS662**

Sixth Semester B.E. Degree Examination, Dec.09/Jan.10

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions.***

1 a. With the help of a neat diagram, explain the various phases of a compiler. **(10 Marks)**

b. Write a transition diagram to recognize the relational operators <, < =, >, > =, < >, =. Write program segments for start state and any one final state. **(10 Marks)**

2 a. With a neat diagram, explain the role of parser. List and explain various error recovery strategies in predictive parsing. **(10 Marks)**

b. Given the grammar
$E \rightarrow EAT/T$, $A \rightarrow + 1 -$, $M \rightarrow *$, $T \rightarrow TMF/F$, $F \rightarrow (E)/num$, do the necessary modifications and construct LL(1) parsing table for the resultant grammar. **(10 Marks)**

3 a. Write an algorithm for constructing an SLR parsing table and explain. **(08 Marks)**

b. Construct LR(O) automation for the grammar $E \rightarrow E + T/T$, $T \rightarrow T * F/F$, $F \rightarrow (E)/id$. **(12 Marks)**

4 a. Construct a canonical LR parsing table for the grammar
$S^1 \rightarrow S$, $S \rightarrow CC$, $C \rightarrow cC$, $C \rightarrow d$. **(14 Marks)**

b. Write the advantages and disadvantages of canonical LR and LALR parsing methods. **(06 Marks)**

5 a. Define synthesized and inherited attributes. Show annotated parse tree for the input 3*5+4n. **(10 Marks)**

b. What is dependency graph? How to construct it for a given parse tree? **(05 Marks)**

c. Describe the methods proposed for evaluating schematic rules. **(05 Marks)**

6 a. Explain in detail, the implementation of three address statements. **(10 Marks)**

b. Give the translation of boolean expressions into three address code. Illustrate the method on inputs if-then and while – do. **(10 Marks)**

7 a. Compare the static and dynamic data storage allocation. **(06 Marks)**

b. Write the structure of an activation record. **(04 Marks)**

c. With examples explain the different parameter passing methods. **(10 Marks)**

8 a. What is basic block? Describe the primary structure preserving transformations on basic blocks. **(10 Marks)**

b. Explain the different issues involved in the design of code generator. **(10 Marks)**

* * * * *

Important Note: 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and/or equations written eg, 42+8=50, will be treated as malpractice.

2

USN □□□□□□□□□□ **06IS662**

Sixth Semester B.E. Degree Examination, May/June 2010

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting at least TWO questions from each part.***

PART – A

1 a. With a neat diagram, explain the different phases of a compiler. **(10 Marks)**

b. Explain the need for multiple passes in compilers. **(06 Marks)**

c. Discuss the various issues of lexical analysis. **(04 Marks)**

2 a. What is left factoring and left recursion elimination? Explain it for the following example:

$E \rightarrow E + T \mid T$

$T \rightarrow id \mid id[\,] \mid id[X]$

$X \rightarrow E, E \mid E$ **(10 Marks)**

b. Construct a transition diagram to accept the set of tokens given below:

Unsigned numbers, identifier **(05 Marks)**

c. For the given grammar:

$S \rightarrow a \mid \uparrow \mid (T)$

$T \rightarrow T, S \mid S$

give leftmost and rightmost derivation for the sentence : (((a, a,), ↑, (a, a)), a) **(05 Marks)**

3 a. Define First and Follow. Determine First and Follow sets for the grammar:

stmt – sequence → stmt stmt – seq′

stmt – seq′ →; stmt – sequence | ∈

stmt → s **(06 Marks)**

b. For the given grammar, construct LL(1) parsing table:

Statement → if – stmt | other

if – stmt →if (exp) statement else – part

else – part → else statement | ∈

exp → 0 | 1 **(08 Marks)**

c. Explain with an example, the stack implementation of shift reduce parser. **(06 Marks)**

4 a. For given grammar, construct SLR(1) parsing table:

$S \rightarrow (S)\,S \mid \in$ **(08 Marks)**

b. For the given grammar, construct LALR(1) parsing table:

$S \rightarrow CC$

$C \rightarrow cC \mid d$ **(12 Marks)**

PART – B

5 a. Briefly explain the concept of syntax directed translation scheme, with an example. **(10 Marks)**

b. For the SDD given below, construct annotated parse tree for the input:

Float a, b, c ;

D → TL ;

T → int | float ;

L → L, id | id **(10 Marks)**

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

6 a. What is dependency graph? Explain with an example. **(08 Marks)**

b. What is Three-address code? List the common Three-address instruction forms. Translate the following into Three-address code:
$S \rightarrow id = E ;$
$E \rightarrow E_1 + E_2 \mid -E_1 \mid (E_1) \mid id$ **(12 Marks)**

7 a. Explain in detail the different storage allocation strategies. **(10 Marks)**

b. Briefly explain the main issues in code generation. **(10 Marks)**

8 a. Explain briefly Dead code elimination. **(06 Marks)**

b. What are basic blocks? For the given intermediate code, generate the flow graph: **(06 Marks)**

```
i = 1
j = 1
t1 = 10 * i
t2 = t1 + j
t3 = 8 * t2
t4 = t3 – 88
a [t4] = 0.0
j = j + 1
if j < = 10 go to (3)
i = i + 1
if i < = 10 go to (2)
i = 1
t5 = i – 1
t6 = 88 * t5
a [t6] = 1.0
i = i + 1
if i < = 10 go to (13)
```

c. Write a note on Peep hole optimization, with an example. **(08 Marks)**

* * * * *

USN □□□□□□□□□□

06CS63

Sixth Semester B.E. Degree Examination, May/June 2010

Compiler Design

Time: 3 hrs. Max. Marks:100

Note: ***Answer any FIVE full questions, selecting at least TWO questions from each part.***

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = , will be treated as malpractice.

PART – A

1 a. Explain, with a neat diagram, the phases of a compiler. **(10 Marks)**

b. Construct a transition diagram for recognizing unsigned numbers. Sketch the program segment to implement it, showing the first two states and one final state. **(10 Marks)**

2 a. Explain the left recursion and show how it is eliminated. Describe the algorithm used for eliminating the left recursion. **(06 Marks)**

b. Eliminate left recursion from the grammar:

$S \rightarrow aB \mid aC \mid Sd \mid Se$

$B \rightarrow bBc \mid f$

$C \rightarrow g$ **(02 Marks)**

c. Given the grammar

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

i) Make necessary changes to make it suitable for LL(1) parsing.

ii) Construct FIRST and FOLLOW sets

iii) Construct the predictive parsing table

iv) Show the moves made by the predictive parser on the input (a, (a, a)) **(12 Marks)**

3 a. Obtain a set of canonical LR(0) items for the grammar: **(08 Marks)**

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow id$

$R \rightarrow L$

b. Is the grammar in Q3(a) SLR(1)? Give reasons. **(04 Marks)**

c. What is handle pruning? Explain with the help of the grammar $S \rightarrow SS + \mid SS* \mid a$ and input string aaa*a++. Give a bottom-up parse of the given input string. **(08 Marks)**

4 a. Given the grammar :

$S \rightarrow AA$

$A \rightarrow Aa \mid b$

i) Construct sets of LR(1) items

ii) Construct canonical LR(1) parsing table. **(12 Marks)**

b. Write a note on the Parser generator – Yacc. **(04 Marks)**

c. Write the Yacc specification of a simple desk calculator with the following grammar for arithmetic expressions,

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid digit$

Where, the token digit is a single digit between 0 and 9. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax-directed definition. **(04 Marks)**

b. Consider the context-free grammar given below: **(08 Marks)**

$S \rightarrow EN$
$E \rightarrow E + T \mid E - T \mid T$
$T \rightarrow T * F \mid T / F \mid F$
$F \rightarrow (E) \mid \text{digit}$
$N \rightarrow ;$

i) Obtain the SDD for the above grammar.

ii) Construct the parse tree, syntax tree and annotated parse tree for the input string 5*6 + 7;

c. Obtain the post-fix SDT for the grammar in Q.5(b) and illustrate the corresponding parser stack implementation. **(08 Marks)**

6 a. Obtain the directed acyclic graph for the expression a + a * (b – c) + (b – c) * d. Also give the sequence of steps for constructing the same. **(06 Marks)**

b. Translate the arithmetic expression a + –(b + c) into quadruples, triples and indirect triples. **(06 Marks)**

c. Explain the syntax-directed translation of switch-statements. **(08 Marks)**

7 a. Describe the general structure of an activation record. Explain the purpose of each item in the activation record. **(06 Marks)**

b. Explain in detail, the strategy for reducing fragmentation in heap memory. **(08 Marks)**

c. Explain briefly the performance metrics to be considered while designing a garbage collector. **(06 Marks)**

8 a. Discuss the issues in the design of a code generator. **(10 Marks)**

b. What are basic blocks and how do you partition a three-address-code into basic blocks? **(05 Marks)**

c. Write the three-address code and construct the basic blocks for the following program segment.

```
Sum = 0;
for (i = 0 ; i < = 10 ; i++)
  Sum = sum + a [i]
```

(05 Marks)

* * * * *