

S.V. National Institute of Technology, Surat
Computer Engineering Department
Mid Semester Examination: M.Sc -III-Mathematics (6th Semester), March -2018
Subject: System Software, Code: CO304

Max Marks: 30

Time: 14:00 to 15:30

Date: 9th march, 2018.

Instructions:

1. Write your B. Tech Admission No/Roll No and other details clearly on the answer books while write your B. Tech Admission No on the question paper, too.
2. Assume any necessary data but give proper justifications.
3. Be precise and clear in answering the questions.

Q1. Answer the following

[12]

1. Generate the variant-I of given assembly language.

[04]

	START	733
	MOVER	CREG, A
	ADD	CREG,=15
	MOVEM	CREG,B
	PRINT	B
LOOP2	MOVER	AREG,='4'
	DIV	AREG, C
LOOP1	COMP	AREG,D
	BC	GE,NXT
	LTORG	
		= '15'
		= '4'
	ORIGIN	LOOP1+10
	ADD	DREG,='15'
	LTORG	
		= '15'
	MOVEM	DREG,RES
	PRINT	ANS-RES
NXT	EQU	LOOP1
	STOP	
A	DC	'20'
B	DS	1
C	DC	'5'
D	DC	'2'
RES	DS	1
	END	

OPCODES:

ADD - 01, SUB-02, MULT - 03, MOVER - 04, MOVEM - 05, COMP-06, BC-07,
 DIV-08, READ - 09, PRINT - 10, STOP - 00. *DS - 02 DC - 01*
 START-01,END-02,ORIGIN-03,EQU-04,LTORG-05. *GE - 05*

[02]

2. Define the following terms
(i) LC Processing
(ii) Back patching
3. Describe following data structures with suitable example: OPTAB, SYMTAB, LITAB and POOLTAB. [03]

OR

How Error handling is performed in Assembly language

4. With proper illustration explain positional parameter and keyword parameter and also show how lexical expansion and semantic expansion differs? [03]
- Q2. Answer the following [18]
1. Discuss the phases of a compiler. [04]
2. Discuss lexical analyzer in brief. [02]
3. Eliminate left recursion from the following grammar: [02]

$S \rightarrow Aa | b$
 $A \rightarrow Ac | Sd | \epsilon$

4. Make the following Grammar Left Factored:
 $S \rightarrow iEs | iEsEs | a$
 $E \rightarrow b$ [02]

5. Consider the following grammar :

$E \rightarrow TE'$

$E' \rightarrow +TE' | \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' | \epsilon$

$F \rightarrow (E) | id$

[06]

- Construct predictive parsing table for the given grammar.
- Show the moves made by predictive parser on input: $id+id*id$.
- Comment on whether the given grammar is LL (1) or not. Justify your answer.

6. Define Recursive Descent Parser and design the algorithm for the following grammar: [02]
- $E \rightarrow iE'$
 $E' \rightarrow +iE' | \epsilon$

Computer Engineering Department, S.V.N.I.T, Surat.
Supplementary Exam – Jan-2018
B.Tech. IIIrd year - VIth Semester
Systems Software (CO304)

Seat No - _____
Total Marks : 50

Date: 30/01/2018

Time : 10:00 to 1:00

Instructions:

1. Write your B.Tech Admission No and other details clearly on the answer book and on the question paper.
2. Assume any necessary data but give proper justifications.
3. Be precise and clear in answering the questions.

Q-1 ANSWER THE FOLLOWING 25

1. Explain Left recursion and Left factoring top down parsing. 5

i. Eliminate left recursion from following grammar.

$S \rightarrow A$

$A \rightarrow Ad \mid Ae \mid aB \mid aC$

$B \rightarrow bBC \mid f$

$C \rightarrow g$

ii. Solve below example by left factoring.

$S \rightarrow \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S \mid a$

$E \rightarrow b$

2. Develop an LL(1) parser table for the following grammar and 5

Parse the string using the parsing table : $(id * id) + (id * id)$

$E \rightarrow TA$

$A \rightarrow +TA \mid \epsilon$

$T \rightarrow VB$

$B \rightarrow *VB \mid \epsilon$

$V \rightarrow id \mid (E)$

3. Construct NFA and DFA for following regular expression: 5

$(0 \mid 1)^* 001 \#$

OR

Construct a DFA without constructing NFA for the following regular expression.

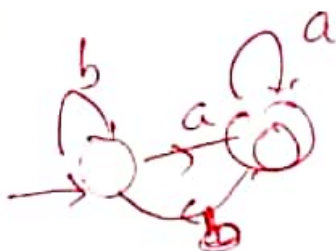
$(a \mid b)^* a$

4. What is peephole optimization? Enlist and explain techniques applied in peephole optimization. 5

5. Construct LR(0) parsing table for the below grammar 5

$S \rightarrow AA$

$A \rightarrow aA/b$



Q-2 ANSWER THE FOLLOWING

25

1. Consider following assembly language program:
Show (i) Contents of Symbol Table (ii) Intermediate codes using Variant 1 representation.

5

AGAIN	START READ MOVER MOVEM MULT MOVER ADD MOVEM COMP BC MOVEM PRINT STOP	101 N BREG, ONE BREG, TERM BREG, TERM CREG, TERM CREG, ONE CREG, TERM CREG, N LE, AGAIN BREG, AGAIN RESULT
N	DS	
RESULT	DS	1
ONE	DC	1
TERM	DS	'1'
	END	1

- Instruction opcode: STOP - 00, ADD - 01, MULT - 03, MOVER - 04, MOVEM - 05, COMP - 06, BC - 07, READ - 09, PRINT - 10, LE - 02 Assembler directives: START - 01, END - 02
- Declaration statements: DC - 01, DS - 02
- Register code: BREG - 02, CREG - 03
2. Define two macros of your choice to illustrate nested calls to these macros. Also show their corresponding expansion. 5
3. Define forward references. How it can be solved using back-patching? Explain with example. 5
4. Explain use and field of following tables of macro. 5
KPDTAB, MDT, EVTAB, SSTAB.
5. Define self-relocating program and write an algorithm of program relocation. 5
- OR
- What is an overlay? Explain overlay structured program and its execution

Sardar Vallabhbhai National Institute of Technology, Surat
Computer Engineering Department
End Semester Examination: M.Sc –III-Mathematics (6th Semester)
Subject: Systems Software, Code: CO304

Max Marks: 50

Time: 12:00 to 15:00

Date: 7th may, 2018.

Instructions:

1. Write your Admission No/Roll No and other details clearly on the answer books while write your Admission No on the question paper, too.
2. Assume any necessary data but give proper justifications.
3. Be precise and clear in answering the questions.

Q. Answer the following

[30]

1. Consider the following grammar :

[08]

$$E \rightarrow E+T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow F$$

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

(a) Construct the collection of sets of LR(0) items for this grammar.

[03]

(b) Construct the parsing table using the SLR algorithm.

[03]

(c) Show all the moves allowed by the parsing table on input $id * id + id$.

[02]

2. Construct the canonical parsing table for the following grammar:

[04]

$$S \rightarrow S$$

$$S \rightarrow CC$$

$$S \rightarrow cC \mid d$$

3. Show that the following grammar is not LALR(1) grammar.

[04]

$$S \rightarrow S$$

$$S \rightarrow aAd \mid bBd \mid aBe \mid aAe$$

$$A \rightarrow c$$

$$B \rightarrow c$$

(a) Define operator grammar. Comment whether the given grammar is operator grammar or not with valid reason. [02] [05]

(b) For the given grammar generate operator-precedence relations and parse the input string : $id+id+id$ [03]

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

5. Compute First and Follow for the following grammar:

[02]

$$S \rightarrow AcB \mid CbB \mid Ba$$

$$A \rightarrow da \mid BC$$

$$B \rightarrow g \mid \epsilon$$

$$C \rightarrow h \mid \epsilon$$

6. Define: handle and handle pruning.
7. (a) Giving a valid reason justify that the following grammar is not suitable for LL (1) parsing.

[02]

- (b) After making it suitable for LL(1) parsing, generate parsing table.

[03]

$S \rightarrow AB$

$A \rightarrow Ca | \epsilon$

$B \rightarrow BaAC | c$

$C \rightarrow b | \epsilon$

Q2. Answer The Following Questions. (Any Two)

1. Define Cross Assembler, Meta Assembler and Micro Assembler.
2. IRP Statement With Suitable Example.
3. What is the significance of object module? List out all the components of object module.

[04]

[02]

[02]

Q3. Answer The Following Questions. (Any Two)

1. Differentiate Non Relocatable programs, Relocatable programs and Self-relocating program. Write an algorithm of program linking.
2. What is an overlay? Explain overlay structured program and its execution.
3. Explain following advanced macro facilities with example:
 1. Alteration of flow of control during expansion
 2. Attributes of formal parameter

[02]

[06]

[03]

[03]

[03]

Q4. Answer The Following Questions.

1. Write a macro which takes A, B as Positional Parameter, C, D as Keyword parameters and calculates $A = A/B + C - D$ in AREG. Also, Generate Data Structures of Macro which you have created.
2. What are the advantages of assembly language over machine level language? Generate Machine Level Language of the given assembly language program.

[10]

[05]

[05]

```

START      200
MOVER      AREG, =5'
MOVEM      AREG, X
MOVER      BREG, =2'
ORIGIN     L1+3
LTORG

L1          -5
           -2
           AREG, =1'
           BREG, =2'
           LT, BACK

NEXT        ADD      -1
           SUB      -2
           BC       EQU
           LTORG    EQU

BACK        EQU      -1
           EQU      -2
           ORIGIN   L1
           MULT     NEXT+5
           STOP     CREG, =4'
           DS       1
           END
  
```

ADD - 01, SUB-02, MULT - 03, MOVER - 04, MOVEM - 05, COMP-06, BC-07, DIV-08, READ - 09, PRINT - 10, STOP - 00.
 START-01, END-02, ORIGIN-03, EQU-04, LTORG-05.
 LT-1.

Dated: 2nd May 2018

Instructions:

1. Write your B Tech Admission No/Roll No and other details clearly on the answer books while write your BTech Admission No on the question paper, too.
2. Assume any necessary data but give proper justifications.
3. Be precise and clear in answering the questions.

Q-1 Explain various phases of a compiler. Describe the output for the following statement, after each phase of a compiler. [04]
 $a = a + b * c * 2.$

Q-2 Answer The Followings: [08]

1. Describe the languages denoted by the following regular expressions: [02]
 - 1) $(aa+ab+ba+bb)^*$
 - 2) $b^*+a^*+(ba)^*$
 - 3) a^*bba^*
 - 4) $((\epsilon/a)b^*)^*$
2. How do the parser and scanner communicate? Explain with block diagram. Also discuss how lexical errors are recovered? [02]
3. What is handle and handle pruning? Explain with example. [02]
4. Eliminate Left recursion for following: [02]
 $A \rightarrow ABd \mid Aa \mid a$
 $B \rightarrow Be \mid b$

Q-3 Answer The Followings: (Any Four) [12]

1. Is the following grammar suitable for LL(1) parsing? If not, make it suitable for it. Compute first and follow. Generate LL(1) parsing table. [03]

$S \rightarrow AB$
 $A \rightarrow Ca \mid \epsilon$
 $B \rightarrow BaAC \mid c$
 $C \rightarrow b \mid \epsilon$

[03]

Check whether the grammar is LR(0), SLR(1)?

$S \rightarrow dA / aB$
 $A \rightarrow bA / c$
 $B \rightarrow bB / c$

Write down functions for non-terminals of the given grammar to implement Recursive decent parser. [03]

$S \rightarrow ABC$
 $A \rightarrow 0A1 \mid \wedge$
 $B \rightarrow 1B \mid \wedge$
 $C \rightarrow 1C0 \mid \wedge$

Consider the following grammar and parse the input string "id+id*id" using operator precedence parser. [03]

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

6. Define: handle and handle pruning.
7. (a) Giving a valid reason justify that the following grammar is not suitable for LL (1) parsing.
- (b) After making it suitable for LL(1) parsing, generate parsing table.

$S \rightarrow AB$
 $A \rightarrow Ca \mid \epsilon$
 $B \rightarrow BaAC \mid c$
 $C \rightarrow b \mid \epsilon$

Q2. Answer The Following Questions. (Any Two)

1. Define Cross Assembler, Meta Assembler and Micro Assembler.
2. IRP Statement With Suitable Example.
3. What is the significance of object module? List out all the components of object module.

Q3. Answer The Following Questions. (Any Two)

1. Differentiate Non Relocatable programs, Relocatable programs and Self-relocating program. Write an algorithm of program linking.
2. What is an overlay? Explain overlay structured program and its execution.
3. Explain following advanced macro facilities with example:
 1. Alteration of flow of control during expansion
 2. Attributes of formal parameter

Q4. Answer The Following Questions.

1. Write a macro which takes A, B as Positional Parameter, C, D as Keyword parameters and calculates $A = A/B + C \cdot D$ in AREG. Also, Generate Data Structures of Macro which you have created.
2. What are the advantages of assembly language over machine level language? Generate Machine Level Language of the given assembly language program.

```

START      200
MOVER      AREG, =5'
MOVEM      AREG, X
MOVER      BREG, =2'
ORIGIN     L1+3
LTORG

L1          -5
           =2
           AREG, =1'
           BREG, =2'
           LT, BACK

NEXT        ADD
            SUB
            BC
            LTORG

BACK        EQU
            ORIGIN
            MULT
            STOP
            DS
            END

X           =1
           =2
           L1
           NEXT+5
           CREG, =4'
           1
  
```

ADD - 01, SUB-02, MULT - 03, MOVER - 04, MOVEM - 05, COMP-06, BC-07, DIV-08, READ - 09, PRINT - 10, STOP - 00.
 START-01, END-02, ORIGIN-03, EQU-04, LTORG-05.
 LT-1.

2. Write Program Relocation Algorithm and Explain Address in Work Area with suitable example. [04]
3. Whether Assembler Need to Produce Program List and Error Reports in PASS-I or Delay This Actions Until PASS-II? Justify your answer with Proper Illustration. [04]

Q-7 Answer The Following Questions: [10]

1. What is main purpose of Object module? Here in this code Three Programs are given in sequence say, Program P, Program Q and Program R. [05]
Generate Object module for each program. While linking program P, Q and R with Link_Origin=799. Display the Content of NTAB.

NOTE: Do not generate machine language program.

Program P	Program Q	Program R
START 121 ENTRY FIRST EXTERN SECOND READA MOVER AREG,A ADD AREG,SECOND SUB AREG,SECOND MOVEM AREG,A A DS 1 FIRST DS 1 END	START 212 ENTRY SECOND EXTERN THIRD,FIRST MOVER AREG,FIRST MULT AREG,THIRD ADD AREG,THIRD MOVEM AREG,X1 X1 DS 1 SECONDD DS 1 END	START 616 ENTRY THIRD EXTERN FIRST,SECOND MOVER AREG,FIRST MULT AREG,SECOND MOVEM AREG,X2 X2 DS 1 THIRD DS 1 END

2. List out The Task Involved in Macro Expansion and Generate the Data Structure of the following Macro. [05]

```

MACRO-
SS2018 &P,&R=DREG,&S=CREG,&U=,&Z=
  LCL &V1
  LCL &V2
&V1 SET 0
&V2 SET 1
  MOVER AREG,&V2
  ADD AREG,&P
  MOVEM AREG,&U
  AIF (&U EQ 10) .L1
  MOVER &S,&Z
  MOVER &R,&U
  MULT &R,&Z
  MOVEM &R,&Z
.L1 MOVEM AREG,&P
  MEND

```

Computer Engineering Department, S V N I T, Surat.
Supplementary Examinations, July 2019
B Tech – III (CO) – 6th semester
Course: (CO-304) Systems Software

Date: 8th July 2019

Time: 10:00 to 13:00

Max Marks: 100

Instructions:

1. Write your B Tech Admission No/Roll No and other details clearly on the answer books and write your B Tech Admission No on the question paper, too.
2. Assume any necessary data but give proper justifications.
3. Be precise and clear in answering the questions.

Q1 Answer the following questions. [60]

1. What is left recursion? Eliminate the left recursion from the following grammar. 6

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

2. Translate the expression $-(a+b)*(c+d)+(a+b+c)$ into 7

1. Quadruples 2. Triples 3. Indirect triples

3. Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment. Indicate corresponding token and pattern. 7

void swap (int a, int b) { int k; k = a; a = b; b = k; }

OR

Construct SLR Parsing Table for the following grammar.

$S \rightarrow OS0 \mid IS1 \mid IO$

4. Construct predictive parsing table for following. If grammar is LL(1) then parse any appropriate string. 8

$S \rightarrow A$

$A \rightarrow aB \mid Ad$

$B \rightarrow bBC \mid f$

$C \rightarrow g$

5. Check whether the following grammar is CLR or not. 8

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

6. Explain Peephole Optimization in detail. 8

7. What are Different Error Recovery Strategies in Compiler? Explain any one with Suitable example. 8
8. What is the difference between parse tree and syntax tree? Write appropriate grammar and draw parse as well as syntax tree for $a+(b*c)$. 8
- Q2. Answer the followings: [Any five] [40]

1. Write a program to allocate sequential block of memory containing first five fibonacci value using explicit expansion time loops facility.
2. Differentiate Non relocatable programs, Relocatable Programs, Self-Relocating Programs.
3. Write an algorithm of program linking.
4. Create two macro of your choice to illustrate the nested call to these macro and show their corresponding expansion.
5. Define the following terms with suitable example:
 - a) Cross Assembler
 - b) Meta Assembler
 - c) Micro Assembler
 - d) AIF statement
6. Generate OPTAB, SYMTAB, LITAB and POOLTAB for the following code.

	START	701		MNEMONICS CODE
	READ	N		STOP 00
	MOVER	BREG,	'=1'	ADD 01
	MOVEM	BREG,	TERM	MULT 03
AGAIN	MULT	BREG,	TERM	MOVER 04
	MOVER	CREG,	TERM	MOVEM 05
	ADD	CREG,	'=1'	COMP 06
	MOVEM	CREG,	TERM	BC 07
	COMP	CREG,	N	READ 09
	BC	LE,	AGAIN	PRINT 10
	MOVEM	BREG,	RESULT	LE 02
	PRINT	RESULT		START 01
	STOP			END 02
N	DS	1		
RESULT	DS	1		
ONE	DC	'1'		
TERM	DS	1		
	END			
	=1			

Supplementary Examination

Computer Engineering Department, S V N I T, Surat.
Supplementary Examination, February 2019

B.Tech (CO) -IIIrd Year semester-VI
Course: Systems Software (CO304)

Dated: 5th FEB 2019

Time: 10:00 to 01:00

Max Marks: 50

Instructions:

1. Write your B Tech Admission No/Roll No and other details clearly on the answer books while write your B.Tech Admission No on the question paper, too.
2. Assume any necessary data but give proper justifications.
3. Be precise and clear in answering the questions.

Q1 Answer The Following Questions.

1. Write a LEX program to count white spaces, tabs and new line. 10
2. Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment. Indicate corresponding token and pattern. 3
3

```
void mult(int i, int j)
{
    int temp;
    temp = i * j;
}
```
3. Test whether the grammar is LL (1) or not, and construct a predictive parsing table for following grammar: 4
 $\begin{aligned} A &\rightarrow AcB \mid cD \mid D \\ B &\rightarrow bB \mid id \\ D &\rightarrow Dab \mid BbB \mid B \end{aligned}$

OR

3. Find out operator precedence relation of the following grammar.
 $\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * X \mid X \\ X &\rightarrow (E) \mid id \end{aligned}$

Q2 Answer The Following Questions.

1. Eliminate Left Recursion from following grammar? 10
 $\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * V \mid V \\ F &\rightarrow id. \end{aligned}$ 3
2. Generate Null NFA (ϵ - NFA) of following Regular Expression 3
 $baa(a|b)^*aab$
3. Construct SLR parsing table for the following grammar: 4
 $\begin{aligned} S &\rightarrow E \\ E &\rightarrow 1 E \\ E &\rightarrow 1 \end{aligned}$

OR

3. Draw a transition diagram for relational operator ($<, <=, =, >, >=, \diamond$).

Q3 Answer The Following Questions.

1. Draw a block diagram of phases of a compiler and indicate the main functions of each phase with an *example. (*take only single example) 10
2. Explain Application domain and Execution domain, and how to bridge the gap between two domains? 3
3. What is shift reduce parser? Consider the following grammar:
 $E \rightarrow E + E, E \rightarrow E * E, E \rightarrow (E), E \rightarrow id$ 3
 Show the shift-reduce parser action for the string $id*(id+id)$. 4

OR

3. What is code optimization? Explain various techniques of code optimization.

Q4 Answer The Following Questions

1. By giving example explain the following term. 5
 1)OPTAB
 2)SYMTAB
 3)LITAB
 4)POOLTAB
2. What is the role of Assembler directives explain any two advanced assembler directives with example. 5

Q5 Answer The Following Questions

1. Differentiate Variant-I and Variant-II and Generate Intermediate codes using Variant I representation. 5

	START	101
	READ	N
	MOVER	BREG, ONE
	MOVEM	BREG, TERM
AGAIN	MULT	BREG, TERM
	MOVER	CREG, TERM
	ADD	CREG, ONE
	MOVEM	CREG, TERM
	COMP	CREG, N
	BC	LE, AGAIN
	MOVEM	BREG, AGAIN
	PRINT	RESULT
	STOP	
N	DS	1
RESULT	DS	1
ONE	DC	'1'
TERM	DS	1
	END	

Instruction opcode: STOP - 00, ADD - 01, MULT - 03, MOVER - 04, MOVEM - 05, COMP - 06, BC - 07, READ - 09, PRINT - 10, LE - 02

Assembler directives: START - 01, END - 02

Declaration statements: DC - 01, DS - 02

Register code: BREG - 02, CREG - 03

2. Enlist and explain task involved in macro expansion 5

OR

In which condition overlay plays an important role? Explain overlay with suitable example.