[This question paper contains 6 printed pages.]

Sr. No. of Question Paper : 7091   F-6   Your Roll No...............

Unique Paper Code   : 2341602

Name of the Paper   : System Programming and Compiler Design

Name of the Course   : **B.Tech. (Computer Science)**

Semester   : VI

Duration : 3 Hours   Maximum Marks : 75

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.

2. The Question paper consists of **two** sections.

3. **Section A** is compulsory.

4. Attempt any **four** questions from **Section B.**

SECTION A

1. (a) List the major data structures used during assembly process. Explain the structure of any one of them. (3)

    (b) Give disadvantages of a single pass assembler over two-pass assembler. (2)

    fwd ref to both data & inst'ns before they are defined
    Some operands are used before they are defined
    Suble errors can not be detected

2. (a) Differentiate between static and shared libraries. (3)
    several progs use single copy of an executable module

    (b) What is the difference between a compiler and an interpreter ? (2)

3. Describe the languages denoted by the following regular expressions

    (a) 0 (0|1)*0 — *the language contains binary no's that starts with a 0 & ends with another 0*

                                        P.T.O.

(b) $((\varepsilon|0)1*)*$ — *All strings of 0's and 1's*

(c) $(0|1)*0(0|1)(0|1)$ — *binary nos that ends with* (1+2+2)

000, 011, 010, 001

4. Write the actions of an LR parse for the following string, for the grammar and parse table shown below:

aa1bbbb

Grammar:

(1) S → A
(2) S → B
(3) A → a A b
(4) A → 0
(5) B → a B bb
(6) B → 1

Handwritten parse trace:

Stack	input	Actions
$0	aa1bbbb$	S1
$0a	a1bbbb$	S1
$0a1a1	1bbbb$	S3
$0a1a113	bbbb$	r6
$0a1a1B7	bbbb$	S9
$0a1a1B7b9	bbb$	S10
$0a1a1B7b9b10	bb$	r5
$0a1B7	bb$	S9
$0a1B7b9	b$	S10
$0a1B7b9b10	$	r5
$0B5	$	r2
$0S1	$	acc

State	a	b	ACTION 0	1	$	GOTO S	A	B
0	S1							
1	S1							
2			S2	S3	$	S		
3			S2	S3		11	4	5
4	r6	r4	r4					7
5	r1	r6	r6					6
6	r2	r1	r6	r4				
7		r2	r1	r6	r6			
8		S8	r2		r1			
9		S9		r2	r2			
10		r3						
11		S10			r5			
		r5			acc			

Symbols used in the above table have their usual meaning.

5. (a) What are the advantages of LALR over Canonical LR parsers? (2)
   smaller tables, of merging, lower memory, does not use back...

   (b) Differentiate between synthesized and inherited attributes. (3)
   get values from child...

---
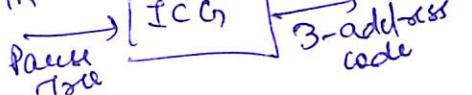
6. What do you mean by Intermediate code generation? List various intermediate code generation schemes. (2+3)

7. Briefly explain the structure of an activation record. (5)

Handwritten activation record diagram: frame pointer, arg n, arg 1 … local variables, return add, temporaries, saved registers, next, arg m, arg n — stack pointer, current frame, prev frame
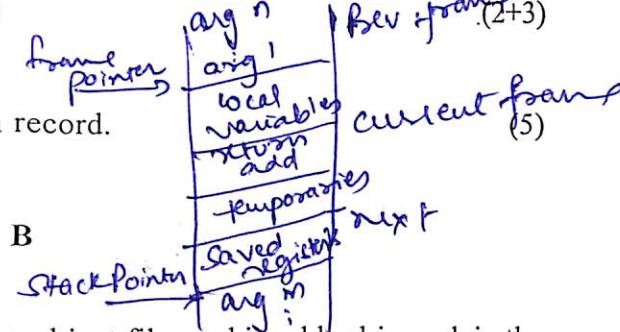
SECTION B

8. (a) Given the following section tables for two object files a.obj and b.obj, explain the linking process and show the layout of the final executable module. (7)

Name	Size	Align
.text	200	16
.data	60	4
.data1	75	4

Section Table for a.obj

Name	Size	Align
.text	100	16
.data	155	4
.data2	300	4

Section Table for b.obj

(b) Briefly explain the Intel hex format for storing object files. (3)
6 fields: Start code, Byte count, Address, Record type, data, Checksum

9. (a) What is relocation? Explain the different ways by which an assembler can pass relocation information to the linker. (3)
link time. 2 Steps: - linker merges all sections in time — by relocation loader of similar types into single section of that type & then assign run time address to each section & symbol.

(b) Assume the following description:
2. each section refers to one or more symbols which are modified to point to correct runtime address based on relocation table

Token	Informal Description
if	charactersi, f
else	characters e, l, s, e
comparison	<or> or <= or >= or == or ! =
id	letter followed by letters and digits
number	A numeric constant
literal	anything but ", surrounded by " 's
Ass_op	=

P.T.O.

Generate the appropriate tokens, lexemes and attribute values for every token that can be generated for the following c++ code fragment :

```
if (x<y)
    c=10;
else
    c=20;
```
(7)

10. Consider the following context-free grammar :

$S \to Sa$

$S \to bS$

$S \to c$

(i) Write the canonical collections of LR(1) items for this grammar. (6)

(ii) Identify all conflicting items, and the types of the conflicts. (4)

11. (a) What are the differences between Panic mode error recovery and Phrase level error recovery in LR parsing ? (3)

(b) Consider the following grammar :

$S \to S S +$

$S \to S S *$

$S \to a$

Show by constructing a SLR parsing table whether the above grammar is SLR or not. (7)

12. (a) Translate the arithmetic expression a*(b+c) into :

(i) Quadruples

(ii) Triples

(iii) Indirect triples (6)

(b) Write type expressions for the following :

(i) A two-dimensional array of integers (i.e., an array of arrays) whose rows are indexed from 0 to 9 and whose columns are indexed from −10 to 10.

(ii) Functions whose domains are functions from integers to pointers to integers and whose ranges are records consisting of an integer and a character. (2+2)

13. (a) Consider the following Syntax directed Definition :

	PRODUCTION	SEMANTIC RULES
1)	$T \to F T'$	$T'.inh = F.val$ $T.val = T'.syn$
2)	$T' \to * F T_1'$	$T_1'.inh = T'.inh \times F.val$ $T'.syn = T_1'.syn$
3)	$T' \to \epsilon$	$T'.syn = T'.inh$
4)	$F \to digit$	$F.val = digit.lexval$

Draw an annotated parse tree for the string : 9*6*7 (4)

(b) Show the stack with all activation record instances, including all the links, when execution reaches position 1 in the following skeletal program. (6)

```
void f3()
{
    ....
}
void f2()
{
    ....
    f3();→ →(1)
    ...
```