

This question paper contains 4+1 printed pages]

Roll No.

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

S. No. of Question Paper : 8837

Unique Paper Code : 234303

C

Name of the Paper : System Programming

Name of the Course : B.Sc. (H) Computer Science Part II

Semester : III

Duration : 3 Hours

Maximum Marks : 75

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

The question paper consists of two Sections.

Section A is compulsory. Attempt any four questions from Section B.

### Section A

1. (a) What is context-sensitive grammar ? Why is it not widely used ? 3
- (b) List the various instruction formats of SIC/XE machine. 4
2. (a) What is the difference between the following :
  - (i) Literal and immediate operand
  - (ii) Compiler and Interpreter. 4
- (b) How should a programmer decide whether to use a macro or subroutine to accomplish a given logical function ? 3
3. (a) What are the advantages and limitations of Linkage editors ? 4
- (b) What is Bootstrap loader ? 3
4. (a) What is the purpose of assembler directives :
  - (i) USE
  - (ii) ORG
  - (iii) RESW
  - (iv) EQU. 4
- (b) State the purpose of lexical, syntactic and semantic analysis. 3

P.T.O.

5. (a) What are P-code compilers ? 3

(b) Consider the following grammar :

<write> → ( <id-list> )

<id-list> → id { , id }

Here, id is a terminal symbol and {} designate repetition of a symbol zero or more times.

Show steps to generate parse tree using recursive descent parser. 4

### Section B

6. (a) State the functions of back end of a language processor. 4

(b) Consider the following code :

	LDS	#3
	LDT	#300
	LDX	#0
ADDLP	LDA	ALPHA, X
	ADD	BETA, X
	STA	GAMMA, X
	ADDR	S, X
	COMPR	X, T
	JLT	ADDLP
	--	
	--	
	--	
ALPHA	RESW	100
BETA	RESW	100
GAMMA	RESW	100

Show the contents of symbol table that would be generated by an assembler. 3

- (c) Give the format of define record, refer record and end record in the object program. 3
7. (a) How is relocation done using modification record ? 2
- (b) Assemble the following SIC source program :

```

SUM      START      4000
FIRST    LDX         ZERO
          LDA         ZERO
LOOP     ADD         TABLE, X
          TIX         COUNT
          JLT         LOOP
          STA         TOTAL
          RSUB
TABLE    RESW        2000
COUNT   RESW        1
ZERO     WORD        0
TOTAL    RESW        1
          END        FIRST

```

Opcodes for the Mnemonics are :

```

LDX      04
LDA      00
ADD      18
TIX      2C
JLT      38
STA      0C
RSUB     4C

```

- (c) Design a Finite Automaton over the alphabet {0,1} which accepts strings starting with 0. 3

8. (a) Consider the following code :

Loc

PROG	START	0
	EXTDEF	A
	EXTREF	B
0020	LDA	A
0023	+LDT	B+4

Write the define record, refer record and modification record that will be generated for the above code. The required relative addresses are given along with the statements. 4

- (b) Describe Automatic Library Search. 3
- (c) Write a Macro to swap two numbers and write a statement to invoke this Macro. 3
9. (a) How are macro instruction parameters concatenated with other character strings ? Describe with an example. 3
- (b) What is a block structured language ? Show how a compiler uses a display data structure for accessing variables in a block. 4
- (c) Which factors determine whether a compiler be designed as one-pass or multi-pass ? 3
10. (a) What is a basic block in a program ? How does it help in code optimization ? 3
- (b) How are literals processed by an assembler ? 3
- (c) Write a LEX program that recognizes positive integers. 4
11. (a) What is program counter relative addressing mode ? Is relocation required in this mode ? Why ? 3

(b) Why the following code cannot be handled by a simple two pass assembler :

```
ALPHA EQU BETA
BETA EQU DELTA
DELTA RESW 1
```

3

(c) Consider the following conditional macro :

```
COND MACRO &A, &B, &C
    IF (&C NE ' ')
        CLEAR A
        CLEAR X
    ENDIF
    IF (&A EQ 1)
        INC A
        RMO A,S
    ENDIF
    IF (&B NE 1)
        COMPR A, S
        JEQ EXIT
    ENDIF
$EXIT STA D
MEND
```

```
D RESW 1
```

Expand the above macro for the following macro calls :

(i) COND 1 2

(ii) COND 2 2 3

4