IS513/IS602

USN	1	M	S							

M S RAMAIAH INSTITUTE OF TECHNOLOGY

(AUTONOMOUS INSTITUTE, AFFILIATED TO VTU)

BANGALORE – 560 054

SEMESTER END EXAMINATIONS – JANUARY 2015

Course & Branch	:	**B.E. – INFORMATION SCIENCE & ENGG.**	**Semester**	:	**V**
Subject	:	**System Software**	**Max. Marks**	:	**100**
Subject Code	:	**IS513/IS602**	**Duration**	:	**3 Hrs**

Instructions to the Candidates:

- Answer one full question from each unit.

UNIT – I

1. a) The variables ALPHA, BETA and GAMMA are arrays of 100 words each. Write an SIC/XE program to add together the corresponding elements of ALPHA and BETA storing the results in the elements of GAMMA. (06)

   b) Suppose FILE is the name of the program written for SIC machine and its length is 0029. Write the object program for the FILE which has the following machine codes: (04)

Loc	Object code
2000	141033
2003	482039
2006	001036
2009	281030
200C	301015
200F	482061
2012	3C1003
2015	00102A
2018	0C1039
201B	05
201C	0C1036
201F	482061
2022	08
2023	4C0000
2026	0C1020

   c) i) Discuss the SIC Machine architecture. (10)
   ii) Provide the format of the modification record. With suitable illustrations indicate why modification records are generated only for format-4 instructions.

2 a) List the functions to be accomplished to translate source program to object code? (05)

   b) Identify the instruction format , addressing mode , program counter or base register relative for the following object codes(in hex) obtained for the SIC/XE instructions and provide suitable explanations. (10)
   1. 69202D
   2. 4B101036
   3. 57C003
   4. B850
   5. 332FFA

c) Illustrate the concept of forward reference with an example. (05)

UNIT – II

3. a) Generate the object code for the following using a 2-pass assembler. Write the block table. (10)

```
            START     0
Start1      LDA    =C'EF'
        STA    BUFF
        +LDT   #MAX
        J      @RET
        USE    CDATA
RET     RESW   1
        LTORG
        USE    CBLKS
BUFF    RESB   4096
BUFE    EQU    *
MAX     EQU    BUFE-BUFF
        END    Start1
```

OPCODES
LDA- 00
STA- 0C
LDT-74
J-3C

b) What is dynamic linking? Describe the process of loading and calling of a subroutine using dynamic linking. (10)

4. a) Explain with an example the concept of load and go one pass assemblers. (06)

b) i. Define absolute loader.Write an algorithm for absolute loaders. (07)

ii. For the given object program explain how relocation is achieved using bit masks.Provide the output of the relocation loaders given that the starting address given by the operating system is 3000.

Input File:

```
H COPY 000000 00
T 000000 C40 141033 481039 901776 921765 345610 571765
T 002011 E00 232838 432979 892060 662849 340000
E 000000
```

c) Illustrate the working of multi-pass assembler for the following sequence of symbol-defining statements that involve forward references. (07)

```
HF EQU  MAX
MAX EQU BU-BUF
BUF RESB 1024
BU  EQU  *
```

Location value for BUF is 1034.

UNIT – III

5. a) Illustrate the generation of unique labels and keyword macro parameters with the examples. (10)

b) Discuss with an example the concept of recursive macro expansion. (06)

c) List any four advantages of general purpose macroprocessor. (04)

6. a) Explain the various data structures used in the implementation of one pass macro processor with an example. (09)

b) Write a short note on macro processing within language translators. (05)

c) Consider a built-in function named %SIZEOF to the macro processor which returns the number of bytes occupied by the corresponding argument. Indicate the expanded code generated by the macroprocessor for the following program. (06)

```
P8 START 0
MOVE MACRO &FROM,&TO
&LENGTH SET %SIZEOF(&FROM)
        IF (&LENGTH EQ 1)
        LDCH &FROM
        STCH  &TO
        ELSE
        LDX  #&LENGTH
        LDS  &FROM
        LDT  &TO
        JSUB  MOVERTN
        ENDIF
        MEND
FIRST  MOVE A,B
        MOVE C,D
        RSUB
A       RESB  1
B       RESB  1
C       RESB  500
D       RESB  500
```

UNIT – IV

7. a) Generate the intermediate code for the following instructions using quadruples. (08)

```
PROGRAM P1
VAR
    M, MQ,inc,value: integer
BEGIN
   M=0; MQ = 5;
   FOR inc:= 10 TO 25 DO
       BEGIN
         READ(value);
         M =M + value;
         MQ = MQ + value;
       END
WRITE (M,MQ);
END.
```

b) Using the grammer given below illustrate a recursive descent parse of the READ(VALUE) statement showing the procedures and a graphic representation of the parsing process. (08)

<read> ::= READ (<id-list>.)
<id-list> :: = id {' , id }

c) Write note on Interpreters. (04)

8. a) Generate the SIC/XE code for the following quadruples. Rearrange them to obtain the SIC/XE optimized code. (08)

```
DIV  V1  #50  i1
*  V2   V2  i2
-   i1   i2  i3
:=  i3       V3
```

b) Illustrate and explain the concept of recursive invocation of a procedure using automatic storage allocation. (08)

c) Illustrate shift reduce parsing. (04)

UNIT – V

9. a) Discuss the following routines used in LEX or YACC programs. (08)
   1. yylex() 2. yywrap() 3. yyparse() 4. yyerror()
   b) Indicate the type of inputs accepted by the following regular expressions. (05)
   1. C{1,3}
   2. -?[0-9]+
   3. 0/1
   4. or|and|but
   5. [a-zA-Z]*
   c) Write an YACC program to recognize a valid C variable. (07)

10. a) Write a lex program to count the number of words in a given input file. (06)

   b) Explain the ambiguity involved while parsing the input 2+3*4*6. Write a YACC program that will handle ambiguity and evaluate the above expression. (10)
   c) With examples describe the terminal and non-terminal symbol used to define a grammer. (04)

****