12

Total Number of Questions : ~~10~~] **[Total Number of pages: 4**

[3663]-255

T.E (I.T.)

SYSTEM SOFTWARE

(2003 COURSE)

Time :3 Hours] [Max Marks : 100

Instructions to candidates :

1) Answer any **THREE** questions from each sections.
2) Answers to the two sections should be written in **separate answer books**.
3) Neat diagrams must be drawn wherever necessary.
4) Figures to the right indicate full marks.
5) Assume suitable data, if necessary.

SECTION – I

Q.1

a) What is language processor development tool? **(02)**

b) Discuss with suitable example Variant-I and Variant-II of intermediate code generation used by the assembler. Also discuss time and space complexity issues. **(08)**

b) What feature of assembly language required us to build a two pass assembler? **(08)**

OR

Q.2

a) With the help of a neat diagram explain the activities involved in Pass-I of a two pass assembler? **(10)**

b) With the help of suitable examples , explain Forward referencing and Back -patching with reference to single pass assembler design. **(08)**

Q.3

a) Can macro-processing be incorporated in an assembler? Justify your answer. (06)

b) Explain following terms with respect to macro: (10)

i) Expansion time variables ii) Conditional assembly iii) Keyword and positional parameters

OR

Q.4

a) What is an open subroutine? Compare macros and subroutines with respect to execution speed and code space requirement.

i) Executing speed ii) Code space requirement (08)

b) Explain two strategies of handling nested macro calls. Compare these two ways with respect to time and space complexity point of view. (08)

Q.5

a) Explain the concept of phases and passes in translators. (08)

b) Consider the following grammar

S→ aAcBe/Ab|b//d

Parse the string "abbcde" using shift reduce parsing technique. (08)

OR

Q.6

a) What is lexical analysis? For the given code, show the output of a lexical analyzer (various tables generated)

```
#include <stdio.h>

main() {

        int i,j;

        for (i=0;i<=10;i++)

        {j=i+10;

        Printf("%d",j); }

        }
```

(08)

b) Generate quadruples for the code given below:

```
 int x[10][10],y[10][10];

for (i = 1;i<=10;i++)

    x [ 1][ 2* i – 1] = y [1][ 2* J];
```

(08)

SECTION --II

Q.7

a) Differentiate between machine dependent and machine independent optimization techniques in compiler. (04)

b) What is the need for generating intermediate code? Explain. (04)

c) With respect to the phases of compiler, state TRUE or FALSE:

i) Memory allocation for an identifier is done by lexical analysis phase

ii) Code generation phase can update Identifier table entries. (02)

d) Discuss the factors affecting target code generation. (08)

OR

Q.8

a) Explain the term activation Record and explain its use in storage allocation. (08)

b) Write simple code generation algorithm. (08)

c) What do you mean by machine independent code optimization. (02)

Q9.

a) What is linkage Editor? What is the essential between linkage editor and linking loader? (08)

b) What would be the advantages and disadvantages of writing a loader using a high level programming language? What problems might you encounter, and how might these be solved? (08)

OR

Q 10

a) Explain Binary Symbolic Subroutines (BSS) loading scheme with example. Also discuss how allocation relocation, linking and loading is done using this scheme. **(10)**

b) Explain the following

i) Dynamic linking   ii) Overlays **(06)**

Q.11

a) Explain various types of loaders. Explain with suitable example overlay structure. **(08)**

b) Explain two pass direct linking loading scheme along with databases / data structure involved in each pass. **(08)**

OR

Q.12

a) Write short notes in:

i) DLL   ii)OLE **(08)**

b) What is clipboard and explain its use in windows programming? **(08)**

xxx