

Duration: 3 hours

Total Marks: 80

- N.B:** (1) Question No. 1 is compulsory.
 (2) Attempt any three questions out of remaining five questions.
 (3) Make suitable assumptions wherever necessary.

- Q.1.** a) Define “System Programming”. Differentiate between system software & application software. [05]
 b) Explain in brief “forward reference problem”. Explain how TII handles forward reference problem in single pass assembler. [05]
 c) Explain conditional macro with suitable example. [05]
 d) Compute FIRST and FOLLOW for the following grammar: [05]
- $$S \rightarrow Aa$$
- $$A \rightarrow BD$$
- $$B \rightarrow b|\epsilon$$
- $$D \rightarrow d|\epsilon$$

- Q.2.** a) Draw the flowchart of pass1 of assembler and explain its working with the databases. [10]
 b) What are the different ways of Intermediate code representation? Explain with example. [10]

- Q.3.** a) Construct the necessary data structures after compiling the following code by Pass1 of two-pass macro processor: [10]

1.	MACRO	
2.	COMPUTE	&x, &a, &p
3.	MOVER	&a, &x
4.	MULT	&a, = '4'
5.	MOVEM	&a, &p
6.	MEND	
7.	MACRO	&g, &k, &r
8.	MOVER	&r, &k
9.	SUB	&r, = '4'
10.	MEND	

- b) Construct LR(0) parsing table for the following grammar and Analyze the contents of stack and input buffer and action taken after each step while parsing the input string “abbcbcdc”: [10]
- $$S \rightarrow aCDe$$
- $$C \rightarrow Cbc$$
- $$C \rightarrow b$$
- $$D \rightarrow d$$

- Q.4.** a) State and explain the types of assembly language statements with examples. [10]
b) Discuss the databases used in direct linking loader. [10]
- Q.5.** a) Generate 3-address code for the following C program and construct flow graph with the help of basic blocks : [10]

```
i=1; j=1; x=5;
while(i<3)
{
    switch(i) {
        case 1: a[j++] = i+x;
                break;
        case 2: a[j++] = i-x;
                break; }
    i++;
}
```

- b) What are the phases of compiler? Give working of each phase for the following statement: [10]
 $P = Q + R - S * 3$
- Q.6.** a) Explain Dynamic Linking Loader in Detail. [10]
b) Explain different Code Optimization Techniques in detail. [10]
