



# Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India  
(Autonomous College Affiliated to University of Mumbai)

## End Semester Examination Synoptic

May 2019

Max. Marks: 60

Class: TE

Course Code: CE61

Name of the Course: (Synoptic) System Programming and Compiler Construction

Duration: 3 Hours

Semester: VI

Branch: Computer/IT

| Question No. |  | Max. Marks |
|--------------|--|------------|
| Q.1 A        | <pre>read first input line if OPCODE='START' then begin     save #[OPERAND] as starting address     initialize LOCCTR to starting address     write line to intermediate file     read next input line end else     initialize LOCCTR to 0 while OPCODE≠'END' do begin     if this is not a comment line then         begin             if there is a symbol in the LABEL field then                 begin                     search SYMTAB for LABEL                     if found then                         set error flag (duplicate symbol)                     else                         insert (LABEL, LOCCTR) into SYMTAB                 end {if symbol}             search OPTAB for OPCODE             if found then                 add 3 {instruction length} to LOCCTR             else if OPCODE='WORD' then                 add 3 to LOCCTR             else if OPCODE='RESW' then                 add 3 * #[OPERAND] to LOCCTR             else if OPCODE='RESB' then                 add #[OPERAND] to LOCCTR             else if OPCODE='BYTE' then                 begin                     find length of constant in bytes                 end         end     end end</pre> | 06         |

```

        add length to LOCCTR
    end {if BYTE}
else
    set error flag (invalid operation code)
end {if not a comment}
write line to intermediate file
read next input line
end {while not END}
Write last line to intermediate file
Save (LOCCTR-starting address) as program length

```

**Steps should be written correctly for finding the value of location counter, filling entries in symbol table and creating opcode table.**

**Skipping of any step or writing incorrect step will account for losing one mark accordingly.  
Every step is indicated correctly, then 6 marks will be given**

|       |  |    |
|-------|--|----|
| Q.1 B | Description of link editor – 3 marks<br>Description of run time linker – 3 marks | 06 |
|-------|--|----|

OR

```

COPY 00100000107A
T0010001E1410334820380010362810393010134820813C100300102A0C103900102D
T003018150C10384820810810334C0000654F48000003000000
T0020394504103000103080205830203810383020573490392C205838203F
T0020571C1010384C00007F10010000410301020793010645090390C20793C1036
T002073073820844C000005
E001000

```

(a) Object program

| Memory address | Contents |          |          |           |
|----------------|----------|----------|----------|-----------|
| 0000           | *****    | *****    | *****    | *****     |
| 0010           | *****    | *****    | *****    | *****     |
| :              | :        | :        | :        | :         |
| 0F00           | *****    | *****    | *****    | *****     |
| 1000           | 14103348 | 20390010 | 38281030 | 30101548  |
| 1010           | 20813C10 | 0300102A | 0C103900 | 102900C10 |
| 1020           | 38482081 | 0810334C | 0000454F | 46000003  |
| 1030           | 000000xx | *****    | *****    | *****     |
|                |          |          |          | COPY      |
| :              | :        | :        | :        | :         |
| 2030           | *****    | *****    | **041030 | 001030E0  |
| 2040           | 205B7020 | 3FD8205D | 18103030 | 26575490  |
| 2050           | 391C205E | 38203F10 | 10364C00 | 00710010  |
| 2060           | 00041030 | E0207930 | 20645090 | 390C2079  |
| 2070           | 2C103638 | 20644C00 | 0005xxxx | *****     |
| 2080           | *****    | *****    | *****    | *****     |
| :              | :        | :        | :        | :         |

(b) Program loaded in memory

Loading of an absolute program.

Design of absolute loader- 3 marks

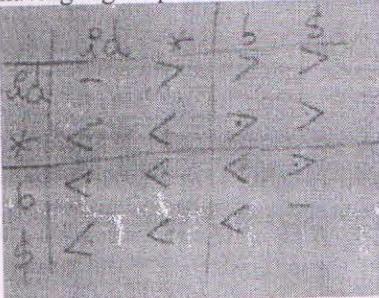
```

begin
    read Header record
    verify program name and length
    read first Text record
    while record type != 'E' do
        begin
            (if object code is in character form, convert into
             internal representation)
            move object code to specified location in memory
            read next object program record
        end
        jump to address specified in End record
    end

```

Algorithm for an absolute loader.

Algorithm of absolute loader – 3 marks

|       |   |    |
|-------|---|----|
| Q.2 A | <p>There are four machine independent macro features:-</p> <ul style="list-style-type: none"> <li>A. Concatenation of macro parameters</li> <li>B. Generation of Unique labels</li> <li>C. Conditional Macro Expansion</li> <li>D. Keyword Macro parameters</li> </ul> <p>Listing – 1 mark<br/>Correct explanation of any one feature – 5 marks</p>   | 06 |
|       |   | 08 |
| Q.2 B | <p>A. Stating correct reason of two adjacent non terminals – 1 mark<br/>Converting into operator grammar – 1 mark<br/>Possible resultant grammar is as follows:</p> <p><math>P \rightarrow SR/S \rightarrow (after substituting values of R) P \rightarrow SbSR/SbS/S \rightarrow (again substituting value of P as SR again) P \rightarrow SbP/SbS/S</math></p> <p>So the operator grammar becomes as follows: (1 mark for converting the grammar in operator grammar)</p> <p><math>P \rightarrow SbP/SbS/S</math><br/> <math>S \rightarrow WbS/W</math><br/> <math>W \rightarrow L^*W/L</math><br/> <math>L \rightarrow id</math></p> <p>Operator Precedence table (3 marks for correct operator precedence table, 0.5 marks will be deducted for each incorrect entry) – id is having higher precedence than all the terminals, * is having higher precedence than b, b and * are right associative.</p>  |    |

Parsing of the input string – 3 marks if clearly shown the contents of Stack, Input and action taken.

OR

4 marks for the correct construction of LR(1) automaton – 1 incorrect state is tolerable, 0.5 marks will be reduced after that.

4 marks for the correct LALR parsing table – 1 incorrect entry in parsing table is tolerable, 0.5 marks will be reduced after that.

|       |  |    |
|-------|--|----|
| Q.3 A | Mentioning types of conflicts- Shift reduce and reduce reduce -1 mark<br>Correct example of shift reduce conflict showing the example entry in the parsing table – 2 marks<br>Correct example of reduce reduce conflict showing the example entry in the parsing table – 2 marks   |    |
| Q.3 B | Keywords like buffer pairs, forward pointer, lexeme begin pointer, sentinels, EOF should be included<br>Algorithm/code for lookahead with sentinels – 2 marks<br>Relevant Explanation with example – 2 marks   | 04 |
| Q.4 A | Writing correct grammar production along with correct SDD with respect to the given example- 2 marks<br>Explanation of the construction of the syntax trees with respect to the mentioned SDD – 4 marks<br>OR<br>Writing correct three address code – 1 mark<br>Explanation of address and register descriptors- 1 mark<br>Correct conversion of three address instruction to assembly instruction and along with that if the contents of the register and address descriptors shown correctly after each step – 4 marks will be allocated, otherwise if incorrect will account of reduction of 1 mark at each step. | 06 |
| Q.4 B | Definition of backpatching – 1 mark<br>Translation scheme for the Boolean expressions – 3 marks<br>Annotated parse tree as per the translation scheme with respect to the given example – 2 marks<br>Three address instructions after filling the labels – 2 marks   | 08 |
| Q.5 A | Correct three address code – 1 mark (Temporary variables should be unique)<br>Quadruple table – 2.5 mark<br>Triple triple- 2.5 marks<br>There should be no mistake in the entries of the tables, otherwise 0.5 marks will be reduced for each incorrect entry.   | 06 |
| Q.5 B | Mark and sweep – 2.5 marks<br>copy collector garbage collection methods – 2.5 marks  | 05 |