

**MODEL ANSWER****SUMMER- 17 EXAMINATION****Subject Title: System Programming**

Subject Code:

**17634****Important Instructions to examiners:**

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.

| Q. No. | Sub Q. N.   | Answer                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Marking Scheme                                        |
|--------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| 1.     |             | <b>Solve any FIVE:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>20Marks</b>                                        |
|        | <b>(1)</b>  | <b>List the four components of system programming.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>4M</b>                                             |
|        | <b>Ans:</b> | <p><b>Assembler:</b> The program known as assembler is written to automate the translation of assembly language to machine language. Input to the language is called as source program and output of assembler is machine language translation called as object program.</p> <p><b>ALP → ASSEMBLER → Machine Language equivalent + Information required by the loader</b></p> <p><b>Loader:</b> Loader is a system program which places program into the memory and prepares for execution. Loading a program involves reading the contents of the executable file containing the program instructions into memory, and then carrying out other required preparatory tasks to prepare the executable for running. Once loading is complete, the operating system starts the program by passing control to the loaded program code. E.g. Boot Strap loader.</p> <p><b>Macro:</b> A macro is a rule or pattern that specifies how a certain input sequence (often a sequence of characters) should be mapped to a replacement output sequence (also often a sequence of characters) according to a defined procedure. The mappings process that instantiates (transforms) a macro use into a specific sequence is known as macro expansion. A facility for writing macros may be provided as part of a software application or as a part of a programming language. In the former case, macros are used to make tasks using the application less repetitive. In the latter case, they are a tool that allows a programmer to enable code reuse or even to design domain-specific languages.</p> | <b>( Each Component: 1 mark, any four Components)</b> |

**MODEL ANSWER****SUMMER- 17 EXAMINATION****Subject Title: System Programming**

Subject Code:

**17634**

MACRO MACRO\_NAME

...

...

...

MEND

**Compiler:** A compiler is a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language, often having a binary form known as object code). The most common reason for converting a source code is to create an executable program. E.g. Javac, TurboC, CC (used in Unix/Linux).

**Linker:** A linker which is also called binder or link editor is a program that combines object modules together to form a program that can be executed. Modules are parts of a program.

**(2)****What are the goals of system software?****4M****Ans:**

Goal of System software

- 1) To achieve efficient use of available resources.
- 2) To achieve efficient performance of the system.
- 3) To make effective execution of general user program.
- 4) To make available new better facilities.
- 5) User convenience - provide convenient methods of using a computer system.
- 6) Non-interference - prevent interference in the activities of its user.

**(Each Goal: 1 mark, any four goals)****(3)****Compare binary search and linear search.****4M****Ans:**

| SR. NO | PARAMETERS                | BINARY SEARCH                                 | LINEAR SEARCH                  |
|--------|---------------------------|-----------------------------------------------|--------------------------------|
| 1      | Time Complexity           | $O(\log_2 N)$                                 | $O(N)$                         |
| 2      | Best case time            | $O(1)$ Center Element                         | $O(1)$ First Element           |
| 3      | Prerequisite for an array | Array must be in sorted order                 | No prerequisite                |
| 4      | Can be implemented on     | Cannot be directly implemented on linked list | Array and Linked list          |
| 5      | Algorithm type            | Divide and conquer in nature                  | Iterative in nature            |
| 6      | Usefulness                | tricky algorithm                              | Easy to use                    |
| 7      | No of Comparison          | More number of comparisons are required       | Number of comparisons are less |

**(Each Comparison Point: 1 mark, any four Comparison points)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| (4)         | <b>Define the terms:</b><br><br><b>i. Allocation      ii. Relocation      iii. Linking      iv. Loading</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>4M</b>                                      |
| <b>Ans:</b> | <p><b>i. Allocation:</b> Allocate the space in the memory where the object programs can be loaded for execution. It allocates the space for program in the memory, by calculating the size of the program. This activity is called allocation.</p> <p><b>ii. Relocation:</b> Adjust the address sensitive instructions to the allocated space. There are some address dependent locations in the program, such address constants must be adjusted according to allocated space, such activity done by loader is called relocation.</p> <p><b>iii. Linking:</b> Resolving external symbol reference. It resolves the symbolic references (code/data) between the object modules by assigning all the user subroutine and library subroutine addresses. This activity is called linking.</p> <p><b>iv. Loading:</b> Placing the object program in the memory in to the allocated space. Finally it places all the machine instructions and data of corresponding programs and subroutines into the memory. Thus program now becomes ready for execution, this activity is called loading.</p>                                                     | <b>(Each term Definition: 1 mark)</b>          |
| (5)         | <b>Explain four basic operations of macroprocessor.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>4M</b>                                      |
| <b>Ans:</b> | <p>The 4 basic task of Macro processor is as follows:-</p> <ol style="list-style-type: none"><li>1) Recognize the macro definitions.</li><li>2) Save the Macro definition.</li><li>3) Recognize the Macro calls.</li><li>4) Perform Macro Expansion.</li></ol> <p>1) <b>Recognize the Macro definitions:</b> - A microprocessor must recognize macro definitions Identified by the MACRO and MEND pseudo-ops. When MACROS and MENDS are nested, the macro processor must recognize the nesting and correctly match the last or outer MEND with the first MACRO.</p> <p>2) <b>Save the Macro definition:</b> - The processor must store the macro instruction definitions which it will need for expanding macro calls.</p> <p>3) <b>Recognize the Macro calls:</b> - The processor must recognize macro call that appear as operation mnemonics. This suggests that macro names be handled as a type of opcode.</p> <p>4) <b>Perform Macro Expansion:</b> - The processor must substitute for macro definition arguments the corresponding arguments from a macro call, the resulting symbolic text is then substituted for the macro call.</p> | <b>(Description of each Operation: 1 mark)</b> |
| (6)         | <b>Describe the term overlay structure.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>4M</b>                                      |
| <b>Ans:</b> | <p>The subroutines of a program are needed at different times. For e.g. Pass 1 and pass 2 of an assembler are call other subroutine it is possible to produce an overlay structure that identifiers mutually exclusive subroutines. In order for the overlay structure to work it is necessary for the module loader to the various procedures as they are</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>(Description: 3 marks, Diagram: 1)</b>      |



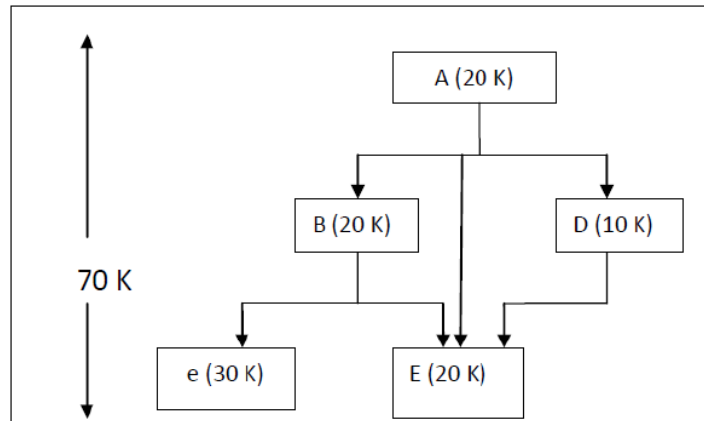
**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

**Subject Code:**

**17634**

needed. The portion of the loader that actually intercepts the “calls” and loads the necessary procedure is collect the overlay supervision or simply the upper.



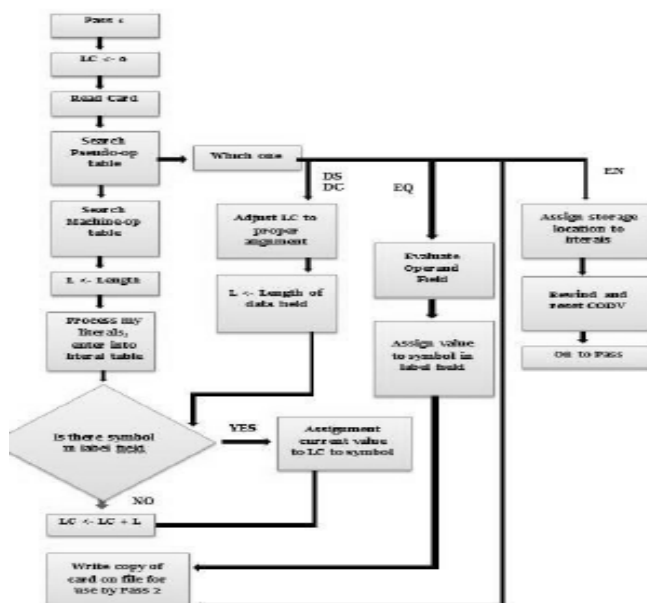
Above program consisting of five subprogram (A, B, C, D & E) that require look bytes of core. The arrow indicate that subprogram A only calls B, D and E; subprogram B only calls C and E; subprogram D only calls E; and subprogram C and E do not call any other routine procedures B and D are never in use at the same time; neither are C and E. If are load only those procedures that are actually to be used at any particular time, the amount of core needed is equal to the longest path of the overlay structure. This happens to be 70k. Overlay reduces the memory requirement of a program. It also makes it possible to execute program where size exceeds the amount of memory which can be allocated to them. For the execution of overlay structured program, the root is loaded in memory and given control for the execution. Other overlays are loaded as and when headed. Loading of an overlay overwrite a previously loaded overlay with the same load origin.

**mark)**

**(7) Draw flow chart of pass-1 of assembler.**

**4M**

**Ans:**



**(Correct flowchart: 4 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

|    |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                      |
|----|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| 2. |             | <b>Solve any FOUR:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <b>16Marks</b>                                       |
|    | <b>(1)</b>  | <b>Describe the concept of compile &amp; go loader.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>4M</b>                                            |
|    | <b>Ans:</b> | <p>One method of performing the loader functions is to have the assembler run in one part of memory and place the assembled machine instructions and data, as they are assembled, directly into their assigned memory locations. As a usual practice one method of performing the loader functions is to have to assemble run in one part of memory and place the assembled machine instructions and data they are assembled, directly into their assigned memory locations. When the assembly is completed the assembler causes transfer to the instruction of the program. This is a simple solution, involving no extra procedures. It is used by the WATFOR FORTRAN compiler and several other language processors. Such a loading scheme is commonly called “compile-and-go” or “assembler – and –go”. It is relatively easy to implement. The assembler simply places the code into core, and the “loader” consists of one instruction that transfers to the starting instruction of the newly assembled program.</p> <p><b>Disadvantages:</b></p> <ol style="list-style-type: none"><li>1. A portion of memory is wasted because the core occupied by the assembler is unavailable to the object program.</li><li>2. It is necessary to retranslate (assemble) the user’s program code every time it is run.</li><li>3. It is very different to handle multiple subroutines in assembly language and another subroutine in any programming language.</li></ol> <p>This last disadvantage makes it very difficult to produce orderly modular programs in the design of assemblers. For example assembler is one of the type of compile and go loader which can be depicted in the following figure:</p> <div data-bbox="324 1423 1307 1787"><p>The diagram illustrates the 'Compile-and-Go' loader process. It shows a flow from 'Source pgn' to 'Compile - and - Go translator'. From the translator, an arrow points to a vertical stack representing memory, which contains 'Program' and 'Assembler' sections. A curved arrow points from the 'Assembler' section back to the 'Compile - and - Go translator'. To the right of the memory stack, a curved arrow labeled 'Control' points towards the 'Program' section. Below the diagram, there is a legend: a bracket on the left side of the memory stack is labeled 'Compile-and-Go', and a bracket on the right side is labeled 'Assemble-and-Go'.</p></div> | <p><b>(Description:3 marks, Diagram: 1 mark)</b></p> |

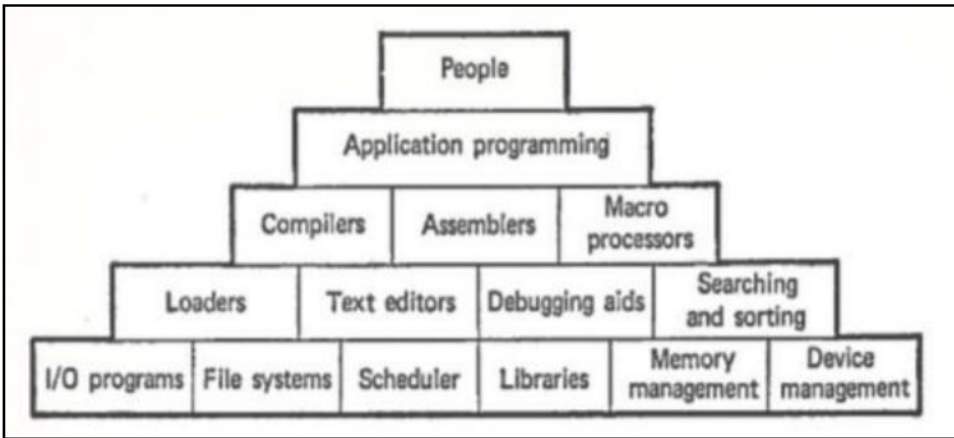
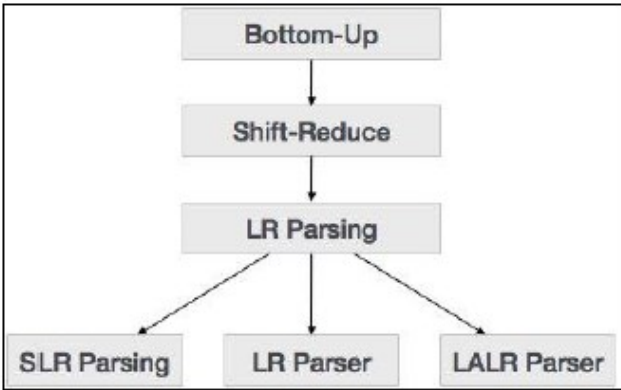


**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                               |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| (2)         | <b>Draw the foundation of system software.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>4M</b>                                     |
| <b>Ans:</b> | <div data-bbox="331 428 1279 863"></div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <b>(Correct Diagram: 4 marks)</b>             |
| (3)         | <b>Explain bottom-up parsing technique.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>4M</b>                                     |
| <b>Ans:</b> | <p data-bbox="232 1035 1349 1178">Bottom-up parsing starts from the leaf nodes of a tree and works in upward direction till it reaches the root node. Here, we start from a sentence and then apply production rules in reverse manner in order to reach the start symbol. The image given below depicts the bottom-up parsers available.</p> <div data-bbox="475 1222 1092 1608"></div> <p data-bbox="232 1656 1349 1728"><b>Shift-Reduce Parsing:</b> Shift-reduce parsing uses two unique steps for bottom-up parsing. These steps are known as shift-step and reduce-step.</p> <p data-bbox="232 1732 1349 1837"><b>Shift step:</b> The shift step refers to the advancement of the input pointer to the next input symbol, which is called the shifted symbol. This symbol is pushed onto the stack. The shifted symbol is treated as a single node of the parse tree.</p> <p data-bbox="232 1841 1349 1980"><b>Reduce step:</b> When the parser finds a complete grammar rule RHS and replaces it to LHS, it is known as reduce-step. This occurs when the top of the stack contains a handle. To reduce, a POP function is performed on the stack which pops off the handle and replaces it with LHS non-terminal symbol.</p> | <b>(Description: 2 Marks, Types: 2 Marks)</b> |

**MODEL ANSWER****SUMMER- 17 EXAMINATION****Subject Title: System Programming**

Subject Code:

**17634**

**LR Parser:** The LR parser is a non-recursive, shift-reduce, bottom-up parser. It uses a wide class of context free grammar which makes it the most efficient syntax analysis technique. LR parsers are also known as LR parsers, where L stands for left-to-right scanning of the input stream; R stands for the construction of right-most derivation in reverse, and k denotes the number of look ahead symbols to make decisions.

There are three widely used algorithms available for constructing an LR parser:

SLR1 – Simple LR Parser:

- Works on smallest class of grammar
- Few number of states, hence very small table
- Simple and fast construction

LR1 – LR Parser:

- Works on complete set of LR1 Grammar
- Generates large table and large number of states
- Slow construction

LALR1 – Look-Ahead LR Parser:

- Works on intermediate size of grammar

Number of states are same as in SLR1

**(4) Explain bucket sort with example.**

**4M****Ans:**

The sort involves examine the least significant digit of the keyword first, and the item is then assigned to a bucket uniquely depend on the value of the digit. After all items have been distributed the buckets items are merged in order and then the process is repeated until no more digits are left. A number system of base P requires P buckets. There are serious disadvantages to using it internally on a digital compiler

1) It takes two separate processes, a separation and a merge

2) It requires a lot of extra Storage for the buckets.

The average time required for the sort is  $(* N * \log P(K))$  where N is the table size, K is the maximum key size & P is the radix of the radix sort. The extra storage required is  $N * P$ .

**(Description:  
2 marks,  
Example:2  
marks)**

| Original table                | First Distribution | Merge                          | First Distribution | Final merge |
|-------------------------------|--------------------|--------------------------------|--------------------|-------------|
| 19                            | 01                 |                                | 01                 |             |
| 13                            | 0)                 | 31                             | 0) 01, 02, 05, 09  | 02          |
| 05                            | 1) 01, 31, 11, 21  | 11                             | 1) 11, 13, 16, 19  | 05          |
| 27                            | 2) 02              | 21                             | 2) 21, 26, 27      | 09          |
| 01                            | 3) 13              | 02                             | 3) 31              | 11          |
| 26                            | 4)                 | 13                             | 4)                 | 13          |
| 31                            | 5) 05              | 05                             | 5)                 | 16          |
| 16                            | 6) 26, 16          | 26                             | 6)                 | 19          |
| 02                            | 7) 27              | 16                             | 7)                 | 21          |
| 09                            | 8)                 | 27                             | 8)                 | 26          |
| 11                            | 9) 19, 09          | 19                             | 9)                 | 27          |
| 21                            | 09                 |                                | 31                 |             |
| ↑                             |                    | ↑                              |                    |             |
| Separate, based on last digit |                    | Separate, based on first digit |                    |             |



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

|  | <b>(5) Explain the concept of subroutine linkages in loader.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>4M</b>                                                                          |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
|  | <p><b>Ans:</b> To understand the concept of subroutine linkages, first consider the following scenario: "In Program "A" call to subroutine B is made. The subroutine B is not written in the program segment of A, rather B is defined in some another program segment C" Nothing is wrong in it. But from assembler's point of view while generating the code for B, as B is not defined in the segment A, the assembler cannot find the value of this symbolic reference and hence it will declare it as an error. To overcome problem, there should be some mechanism by which the assembler should be explicitly informed that segment B is really defined in some other segment C. Therefore whenever segment B is used in segment A and if at all B is defined in C, then B must -be declared as an external routine in A. To declare such subroutine as external, we can use the assembler directive EXT. Thus the statement such as EXT B should be added at the beginning of the segment A. This actually helps to inform assembler that B is defined somewhere else. Similarly, if one subroutine or a variable is defined in the current segment and can be referred by other segments then those should be declared by using pseudo-ops INT. Thereby the assembler could inform loader that these are the subroutines or variables used by other segments. This overall process of establishing the relations between the subroutines can be conceptually called a _ subroutine linkage.</p> <p>For example<br/>MAIN START<br/>EXT B<br/>.<br/>.<br/>.<br/>CALL B<br/>.<br/>.<br/>END<br/>B START<br/>.<br/>.<br/>RET<br/>END</p> <p>At the beginning of the MAIN the subroutine B is declared as external. When a call to subroutine B is made, before making the unconditional jump, the current content of the program counter should be stored in the system stack maintained internally. Similarly while returning from the subroutine B (at RET) the pop is performed to restore the program counter of caller routine with the address of next instruction to be executed.</p> | <p><b>(Description: 4 marks, any relevant description shall be considered)</b></p> |



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

**Subject Code:**

**17634**

| (6)         | <b>Draw &amp; explain passes of compiler.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <b>4M</b>                                                                |         |           |   |                                                                                             |                                                                          |   |                                         |                                  |                             |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|---------|-----------|---|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|---|-----------------------------------------|----------------------------------|-----------------------------|
| <b>Ans:</b> | <div data-bbox="380 401 1187 852"></div> <p><b>Passes is a logical execution of the compilation process.</b></p> <ul style="list-style-type: none"><li>i. <b>Pass 1</b> This pass is corresponds to lexical analysis phase. This pass scans the source code and creates an identifier, literal and uniform symbol table.</li><li>ii. <b>Pass 2</b> corresponds to syntactic and interpretation phase. It scans the uniform symbol table, produces the matrix and place information about identifier into the identifier table.</li><li>iii. <b>Passes 3 to N-3</b> corresponds to the optimization phase. Each separate type of optimization may require several passes over the matrix.</li><li>iv. <b>Pass N-2</b> corresponds to the storage assignment phase. This is a pass over the identifier and literal tables rather than program itself.</li><li>v. <b>Pass N-1</b> corresponds to the code generation phase. It scans the matrix and creates the first version of the object deck.</li><li>vi. <b>Pass N</b> corresponds to the assembly phase. It resolves the symbolic addresses and creates information for the loader.</li></ul> | <b>(Diagram: 2 marks,<br/>Description: 2 marks)</b>                      |         |           |   |                                                                                             |                                                                          |   |                                         |                                  |                             |
| <b>3.</b>   | <b>Solve any FOUR:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>16Marks</b>                                                           |         |           |   |                                                                                             |                                                                          |   |                                         |                                  |                             |
| (1)         | <b>Differentiate between searching and sorting.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>4M</b>                                                                |         |           |   |                                                                                             |                                                                          |   |                                         |                                  |                             |
| <b>Ans:</b> | <table><tr><th data-bbox="224 1581 305 1675">Sr. no</th><th data-bbox="305 1581 797 1675">Sorting</th><th data-bbox="797 1581 1344 1675">Searching</th></tr><tr><td data-bbox="224 1675 305 1822">1</td><td data-bbox="305 1675 797 1822">It is a process of arranging collection of items in ascending order or in descending order.</td><td data-bbox="797 1675 1344 1822">It is the process of finding a data element in the given data structure.</td></tr><tr><td data-bbox="224 1822 305 1948">2</td><td data-bbox="305 1822 797 1948">To arranging data in some logical order</td><td data-bbox="797 1822 1344 1948">To check existence of an element</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Sr. no                                                                   | Sorting | Searching | 1 | It is a process of arranging collection of items in ascending order or in descending order. | It is the process of finding a data element in the given data structure. | 2 | To arranging data in some logical order | To check existence of an element | <b>(Each Point: 1 mark)</b> |
| Sr. no      | Sorting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Searching                                                                |         |           |   |                                                                                             |                                                                          |   |                                         |                                  |                             |
| 1           | It is a process of arranging collection of items in ascending order or in descending order.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | It is the process of finding a data element in the given data structure. |         |           |   |                                                                                             |                                                                          |   |                                         |                                  |                             |
| 2           | To arranging data in some logical order                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | To check existence of an element                                         |         |           |   |                                                                                             |                                                                          |   |                                         |                                  |                             |



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

|         |                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|------------------------|-----------------------------------------------------------|-------------------------------------------|------|-------|-------|---------------------------------------------------------------------------------------|-------|--------------------------|----|----------------------|-------|---------------------------|----|-------|-------|-----------------------------------|-------|-------|-------|----------------------|----------------------|
|         |                                                                                                                                                                          | <table><tr><td>3</td><td>It is used to arrange the data items in some order i.e. in ascending or descending order in case of numerical data and in dictionary order in case of alphanumeric data.</td><td>To determine position of a search element and to determine number of occurrences of a search element.</td></tr><tr><td>4</td><td>Example:- Radix sort, Bubble sort, Radix interchange sort</td><td>Example:- Linear search and binary search</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 3       | It is used to arrange the data items in some order i.e. in ascending or descending order in case of numerical data and in dictionary order in case of alphanumeric data. | To determine position of a search element and to determine number of occurrences of a search element. | 4                      | Example:- Radix sort, Bubble sort, Radix interchange sort | Example:- Linear search and binary search |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 3       | It is used to arrange the data items in some order i.e. in ascending or descending order in case of numerical data and in dictionary order in case of alphanumeric data. | To determine position of a search element and to determine number of occurrences of a search element.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 4       | Example:- Radix sort, Bubble sort, Radix interchange sort                                                                                                                | Example:- Linear search and binary search                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
|         | (2)                                                                                                                                                                      | What type of information is contained by ESD, RLD, TXT, END cards of direct linking loader?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 4M      |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
|         | Ans:                                                                                                                                                                     | <p>There are four sections of the object deck for a direct linking loader. The ESD card the information necessary to build the external symbol. The external symbols are symbols that can be referred beyond the subroutine level. The normal labels in the source program are used only by the assembler. The ESD card contains the information necessary to build the external symbol. The external symbols are symbols that can be referred beyond the subroutine level. The normal labels in the source program are used only by the assembler.</p> <p><b>ESD card format:</b></p> <table><tr><td>Columns</td><td>Contents</td></tr><tr><td>1</td><td>Hexadecimal byte X'02'</td></tr><tr><td>2-4</td><td>Characters ESD</td></tr><tr><td>5-14</td><td>Blank</td></tr><tr><td>15-16</td><td>ESD identifier (ID) for program name (SD) external symbol(ER) or blank for entry (LD)</td></tr><tr><td>17-24</td><td>Name, padded with blanks</td></tr><tr><td>25</td><td>ESD type code (TYPE)</td></tr><tr><td>26-28</td><td>Relative address or blank</td></tr><tr><td>29</td><td>Blank</td></tr><tr><td>30-32</td><td>Length of program otherwise blank</td></tr><tr><td>33-72</td><td>Blank</td></tr><tr><td>73-80</td><td>Card sequence number</td></tr></table> <p><b>TXT card:</b></p> <p>The TXT card contains the blocks of data and the relative address at which data is to be placed. Once the loader has decided where to load the program, it adds the Program Load Address (PLA) to relative address. The data on the TXT card may be instruction, non-related data or initial values of address constants.</p> | Columns | Contents                                                                                                                                                                 | 1                                                                                                     | Hexadecimal byte X'02' | 2-4                                                       | Characters ESD                            | 5-14 | Blank | 15-16 | ESD identifier (ID) for program name (SD) external symbol(ER) or blank for entry (LD) | 17-24 | Name, padded with blanks | 25 | ESD type code (TYPE) | 26-28 | Relative address or blank | 29 | Blank | 30-32 | Length of program otherwise blank | 33-72 | Blank | 73-80 | Card sequence number | (Each Card: 1 mark ) |
| Columns | Contents                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 1       | Hexadecimal byte X'02'                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 2-4     | Characters ESD                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 5-14    | Blank                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 15-16   | ESD identifier (ID) for program name (SD) external symbol(ER) or blank for entry (LD)                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 17-24   | Name, padded with blanks                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 25      | ESD type code (TYPE)                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 26-28   | Relative address or blank                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 29      | Blank                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 30-32   | Length of program otherwise blank                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 33-72   | Blank                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |
| 73-80   | Card sequence number                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                                                                                                                                                                          |                                                                                                       |                        |                                                           |                                           |      |       |       |                                                                                       |       |                          |    |                      |       |                           |    |       |       |                                   |       |       |       |                      |                      |

**MODEL ANSWER****SUMMER- 17 EXAMINATION****Subject Title: System Programming**

Subject Code:

**17634****format**

| Columns | Contents                                                      |
|---------|---------------------------------------------------------------|
| 1       | Hexadecimal byte X'02'                                        |
| 2-4     | Characters TXT                                                |
| 5       | Blank                                                         |
| 6-8     | Relative address of first data byte                           |
| 9-10    | Blanks                                                        |
| 11-12   | Byte Count (BC) = number of bytes of information in cc. 17-72 |
| 13-16   | Blank                                                         |
| 17-72   | From 1 to 56 data bytes                                       |
| 73-80   | Card sequence number                                          |

**RLD card**

The RLD cards contain the following information 1. The location and length of each address constant that needs to be changed for relocation or linking. 2. The external symbol by which the address constant should be modified. 3. The operation to be performed.

**RLD card format:**

| Columns | Contents                                                      |
|---------|---------------------------------------------------------------|
| 1       | Hexadecimal byte X'02'                                        |
| 2-4     | Characters RLD                                                |
| 5-18    | Blank                                                         |
| 19-20   | Relative address of first data byte                           |
| 21      | Blanks                                                        |
| 22-24   | Byte Count (BC) = number of bytes of information in cc. 17-72 |
| 25-72   | Blank                                                         |
| 73-80   | Card sequence number                                          |

**END card**

The END card specifies the end of the object deck.

**END card format:**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

|         |                                                                     | <table><tr><th>Columns</th><th>Contents</th></tr><tr><td>1</td><td>Hexadecimal byte X'02'</td></tr><tr><td>2-4</td><td>Characters END</td></tr><tr><td>5</td><td>Blank</td></tr><tr><td>6-8</td><td>Start of execution entry (ADDR), if other than beginning of program</td></tr><tr><td>9-72</td><td>Blanks</td></tr><tr><td>73-80</td><td>Card sequence number</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Columns                    | Contents | 1 | Hexadecimal byte X'02' | 2-4 | Characters END | 5 | Blank | 6-8 | Start of execution entry (ADDR), if other than beginning of program | 9-72 | Blanks | 73-80 | Card sequence number |  |
|---------|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|----------|---|------------------------|-----|----------------|---|-------|-----|---------------------------------------------------------------------|------|--------|-------|----------------------|--|
| Columns | Contents                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                            |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
| 1       | Hexadecimal byte X'02'                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                            |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
| 2-4     | Characters END                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                            |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
| 5       | Blank                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                            |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
| 6-8     | Start of execution entry (ADDR), if other than beginning of program |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                            |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
| 9-72    | Blanks                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                            |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
| 73-80   | Card sequence number                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                            |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
|         | (3)                                                                 | State four functions of compiler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 4M                         |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
|         | Ans:                                                                | 1. Recognizing basic elements like variable, keyword, operators.<br>2. Recognizing combination of elements as a syntactic unit and interpret their meaning<br>3. Storage allocation for variables<br>4. Generate the equivalent object code.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | ( Four Functions: 4 marks) |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
|         | (4)                                                                 | Define the terms:<br><br>(i)Binder (ii). Dynamic loader (iii). Linking editor (iv). Overlays.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 4M                         |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
|         | Ans:                                                                | <p><b>1. Binders:</b> A binder is a program that performs the same functions as the direct-linking loader in “binding” subroutines together, but rather than placing the relocated and linked text directly into memory, it outputs the text as a file or card deck. This output file is in a format ready to be loaded and is typically called a load module.</p> <p><b>2) Dynamic Loader:</b> The portion of the loader that actually intercepts the “calls” and loads the necessary procedure is called the overlay supervisor or simply the flipper. This overall scheme is called dynamic loading – on call (LOCAL).</p> <p><b>3) Linkage Editor:</b> A more sophisticated binder, called a linkage editor, can keep track of then relocation information so that the resulting load module, as an ensemble, can be further relocated and thereby loaded anywhere in core.</p> <p><b>4) Overlay:</b> An overlay is a part of a program (or software package) which has the same load origin as some other part(s) of the program. Overlays are used to reduce the main memory requirement of a program.</p> | (Each Term: 1 mark)        |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
|         | (5)                                                                 | Write algorithm for syntax analysis phase of compiler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 4M                         |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |
|         | Ans:                                                                | Syntax analysis Phase: Algorithm<br>1. Reduction are checked repeatedly for match between the field which old top of stack and the actual top of stack , until match is found<br>2. If match is found, action routines specified in the action field are executed.<br>3. After syntax analyzer get control back to it, it modifies the top of stack with new top                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | ( Four Steps: 4 marks )    |          |   |                        |     |                |   |       |     |                                                                     |      |        |       |                      |  |

**MODEL ANSWER****SUMMER- 17 EXAMINATION****Subject Title: System Programming**

Subject Code:

**17634**

of stack

4. Step 1 is repeated starting with the reduction specified in the next reduction field.

**(6)****Explain the structure of Macro Definition Table (MDT) and Argument List Array (ALA) with example.****4M****Ans:**

MDT: - MDT is used to save the macro definition along with MEND statement. Each parameter is replaced by the index notation. In pass 1 MDT is used to save the macro definition. In pass 2 MDT is used for performing macro expansion

**Structure of MDT:**

| INDEX | MACRO DEFINITION TABLE 80 BYTES/ENTRY |
|-------|---------------------------------------|
|       |                                       |
|       |                                       |

ALA:-It is used for simplifying the parameter replacement procedure. In pass 1 ALA is used to replace the formal parameter by three respective index notations In pass 1 ALA is used to replace the index notations by three actual parameters.

Dummy arg in the macro definition are placed with positional indicators when stored in MDT

**Structure of ALA:**

# symbol is used as index marker ,ith dummy arg represented by #i

| INDEX | ARGUMENT (8 BYTES PER ENTRY) |
|-------|------------------------------|
|       |                              |
|       |                              |

Example:-

Source code

Macro

&amp;LAB INCR &amp;A1,&amp;A2,&amp;A3

&amp;LAB A 1,&amp;A1

A 1,&amp;A2

A 1, &amp;A3

MEND

**(MDT with Example: 2 marks, ALA with Example: 2 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

MNT

| INDEX | NAME     | MDT POINTER(4 BYES) |
|-------|----------|---------------------|
| .     | .        | .                   |
| .     | .        | .                   |
| .     | .        | .                   |
| 3     | INCRbbbb | 15                  |
| .     | .        | .                   |
| .     | .        | .                   |
| .     | .        | .                   |

Pass 2 (ALA)

LOOP INCR DATA1,DATA2,DATA3

| INDEX | ARGUMENT(8 BYTES PER ENTRY) |
|-------|-----------------------------|
| 0     | "LOOPbbbb"                  |
| 1     | "DATA1bbb"                  |
| 2     | "DATA2bbb"                  |
| 3     | "DATA3bbb"                  |

b denotes :- blank

**MDT:**

| INDEX | INDEX MACRO<br>DEFINITION TABLE 80<br>BYTES/ENTRY |
|-------|---------------------------------------------------|
| 15    | &LAB INCR &A1,&A2,&A3                             |
| 16    | #0 A 1,#1                                         |
| 17    | A 1,#2                                            |
| 18    | A 1,#3                                            |
| 19    | MEND                                              |

**4.**

**Solve any FOUR:**

**16Marks**

**(1)**

**Explain working of relocating loader.**

**4M**

**Ans:**

Loader avoids possible reassembling of all subroutines when a single subroutine is changed and perform the tasks of allocation and linking for the programmer. The BSS loader allows many procedure segments yet only one data segment. The assembler assembles each procedure segment independently and passes on to the loader the text and information as to relocation and inter segment references. The o/p of a relocating

**(Working: 4 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

assembler using a BSS scheme is the object program and information about all other program it references. For each source program the assembler o/p a text prefixed by a transfer vector that consist of addresses containing names of the subroutines referenced by the source program. The assembler would also provide the loader with additional information such as the length of the entire program and the length of the transfer vector position. After loading the text and the transfer vector into core, the loader would load each subroutine identified in the transfer vector. It would the place a transfer instruction to the corresponding subroutine in each entry in the transfer vector. The BSS loader scheme is other used on compiler with a fixed length direct address instruction format.

The relocation bit solves the problem of relocation, the transfer vector is used to solve the problem of linking and the program length information solves the problem of allocation.

**(2) Draw format of MOT, POT, ST, BT database of pass-1 of assembler.**

**4M**

**Ans: 1. Machine Op-code Table (MOT):-**

| Mnemonic<br>opcode 4 Bytes | Binary Opcode<br>1 Byte | Length 2<br>Bit | Instruction<br>Format 3 Bit | Reserved 3Bit |
|----------------------------|-------------------------|-----------------|-----------------------------|---------------|
|                            |                         |                 |                             |               |

**Pseudo-Opcode Table (POT):-**

| Pseudo-Opcode 5 Bytes | Address of Function 3 Bytes<br>=24 bit address |
|-----------------------|------------------------------------------------|
|                       |                                                |

**Symbol Table(ST) :-**

| Symbol Table 8 Bytes | Value 4 Bytes | Length 1 Bytes | Relocation 1 Bytes |
|----------------------|---------------|----------------|--------------------|
|                      |               |                |                    |

**Base Table (BT):-**

**(Each  
Database  
format: 1  
mark)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

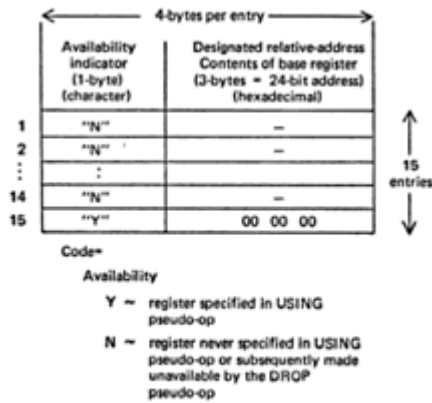
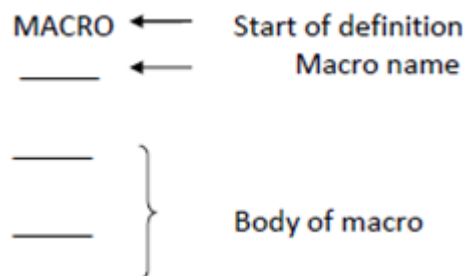


FIGURE 3.9 Base Table (BT) for pass 2

(3) Draw macro instruction structure.

4M

Ans:



(Structure:  
4 marks )



**MODEL ANSWER**

**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

—————  
MEND ← End of macro

E.g:-

```

MACRO
INCR
A1, DATA
A2, DATA
A3, DATA
MEND
.
.
.
INCR
.
.
.
END

```

**(4) Explain the structure of:**

**i. Identifier Table    ii. Matrix Database of compiler.**

**4M**

**Ans:** **IDENTIFIER TABLE:** Created by lexical analysis to describe all identifiers used in the source program. There is one entry for each identifier. Lexical analysis creates the entry and places the name of the identifier into that entry. Since in many languages identifiers may be from 1 to 31 symbols long. The lexical phase may enter a pointer in the identifier table for efficiency of storage. The pointer points to the name in a table of names. Later phases will fill in the data attributes and address of each identifier.

| Name | Data Attributes | Address |
|------|-----------------|---------|
|      |                 |         |

**MATRIX:** The compiler uses the matrix, which is the intermediate form of a linear representation of the past tree. Each matrix entry has one operator and two operands and intermediate result store in temporary variable.

**(Identifier Table : 2 marks, Matrix Database table: 2 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

|                |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                         |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|----------|-----------|-----------|--------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|                |          | <table><tr><td>MATRIX LINE NO</td><td>OPERATOR</td><td>OPERAND 1</td><td>OPERAND 2</td><td>RESULT</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | MATRIX LINE NO                          | OPERATOR | OPERAND 1 | OPERAND 2 | RESULT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| MATRIX LINE NO | OPERATOR | OPERAND 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | OPERAND 2                               | RESULT   |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                         |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                         |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                         |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                | (5)      | Explain Random Entry Searching.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 4M                                      |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                | Ans:     | <p>All Binary Search algorithms, which are fast, but can only operate on tables that are ordered and packed, i.e. tables that will have adjacent items ordered by keywords. Such search procedures may therefore have to be used in conjunction with a sort algorithm which both orders and packs the data. Actually, it is unnecessary for the table to be ordered and packed to achieve good speed in searching. This is also possible to do considerably better with an unpacked, unordered table, provided it is sparse, i.e. the number of storage spaces allocated to it exceeds the number of items to be stored. It is observed that the address calculation sort gives good results with a sparse table. However, having to put elements in order slows down the process. A considerable improvement can be achieved by inserting element in a random (or pseudo-random) way. The random entry number K is generated from the key by methods similar to those used in address calculation. If the K the some other cell must be found for the insertion. The first problem is the generation of a random number from the key. It is to design a procedure that will generate pseudo-random, consistent table positions for keywords. One fairly good prospect for four character EBCDIC keywords is to simply divide the keyword by the table length N and use the remainder. This scheme works well as long as N and the key size (32 bits in case) have no common factors. For a given group of M keywords the remainders should be fairly evenly distributed over).... (N-1).</p> | (Explanation: 4 marks)                  |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                | (6)      | Explain Machine dependent optimization phase of compiler with example.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 4M                                      |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|                | Ans:     | <p>Machine dependent optimization:</p> <ul style="list-style-type: none"><li>● If we optimize register usage in the matrix, it becomes machine – dependent optimization.</li><li>● Following figure depicts the matrix that we previously optimized by eliminating a common sub expression (M4).</li><li>● Next to each matrix entry is a code generated using the operators.</li><li>● The third column is even better code in that it uses less storage and is faster due to a more appropriate mix of instructions.</li><li>● This example of machine-dependent optimization has reduced both the memory space needed for the program and the execution time of the object program by a factor of 2.</li><li>● Machine dependent optimization is typically done while generating code</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | (Description:2 marks, Example: 2 marks) |          |           |           |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

| Optimized Matrix |   |       |        | First try |            | Improved code |            |    |   |    |
|------------------|---|-------|--------|-----------|------------|---------------|------------|----|---|----|
|                  |   |       |        | L         | 1, START   | L             | 1, START   |    |   |    |
| 1                | - | START | FINISH | S         | 1, FINISH  | S             | 1, FINISH  | M1 | → | R1 |
|                  |   |       |        | ST        | 1, M1      |               |            |    |   |    |
|                  |   |       |        |           |            |               |            |    |   |    |
|                  |   |       |        | L         | 1, RATE    | L             | 3, RATE    |    |   |    |
| 2                | * | RATE  | M1     | M         | 0, M1      | MR            | 2, 1       | M2 | → | R3 |
|                  |   |       |        | ST        | 1, M2      |               |            |    |   |    |
|                  |   |       |        |           |            |               |            |    |   |    |
|                  |   |       |        | L         | 1, =F'2'   | L             | 5, = F'2'  |    |   |    |
| 3                | * | 2     | RATE   | M         | 0, RATE    | M             | 4, RATE    | M3 | → | R5 |
|                  |   |       |        | ST        | 1, M3      |               |            |    |   |    |
|                  |   |       |        |           |            |               |            |    |   |    |
| 4                |   |       |        |           |            |               |            |    |   |    |
|                  |   |       |        |           |            |               |            |    |   |    |
|                  |   |       |        | L         | 1, M1      |               |            |    |   |    |
| 5                | - | M1    | 100    | S         | 1, =F'100' | S             | 1, =F'100' | M5 | → | R1 |
|                  |   |       |        | ST        | 1, M5      |               |            |    |   |    |
|                  |   |       |        |           |            |               |            |    |   |    |
|                  |   |       |        | L         | 1, M3      | LR            | 7, 5       |    |   |    |
| 6                | * | M3    | M5     | M         | 0, M5      | MR            | 6, 1       | M6 | → | R7 |
|                  |   |       |        | ST        | 1, M6      |               |            |    |   |    |
|                  |   |       |        |           |            |               |            |    |   |    |
|                  |   |       |        | L         | 1, M2      |               |            |    |   |    |
| 7                | + | M2    | M6     | A         | 1, M6      | AR            | 3, 7       | M7 | → | R3 |
|                  |   |       |        |           | 1, M7      |               |            |    |   |    |
|                  |   |       |        | L         | 1, M7      | ST            | 3, COST    |    |   |    |
|                  |   |       |        |           |            |               |            |    |   |    |
| 8                |   | M7    | COST   | ST        | 1, COST    |               |            |    |   |    |

|    |             |                                                                                                                                                                                                                                                                                                                                                |                               |
|----|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| 5. |             | <b>Solve any FOUR:</b>                                                                                                                                                                                                                                                                                                                         | <b>16Marks</b>                |
|    | (1)         | <b>Explain macro within macro.</b>                                                                                                                                                                                                                                                                                                             | <b>4M</b>                     |
|    | <b>Ans:</b> | <p>Macro calls are “abbreviations” of instruction sequences, it seems reasonable that such “abbreviations” should be available within other macro definitions.</p> <p>For example,</p> <pre> MACRO ADD1 &amp;ARG L 1, &amp;ARG A 1, =F'1" ST 1, &amp;ARG MEND  MACRO ADDS &amp;ARG1, &amp;ARG2, &amp;ARG3 ADD1 &amp;ARG1 ADD1 &amp;ARG2 </pre> | <b>(Explanation: 4 marks)</b> |



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

ADD1 &ARG3  
MEND

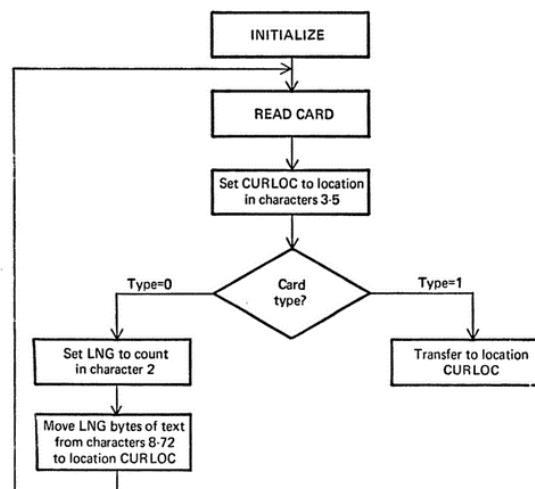
| Source                                                                             | Expanded source (level 1)                                          | Expanded source (level 2)                                                                                                                            |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| MACRO ADD1 &ARG<br>L 1,&ARG<br>A 1,=F '1'<br>ST 1,&ARG<br>MEND                     |                                                                    |                                                                                                                                                      |
| MACRO ADDS &ARG1,<br>&ARG2,&ARG3<br>ADD1 &ARG1<br>ADD1 &ARG2<br>ADD1 &ARG3<br>MEND |                                                                    |                                                                                                                                                      |
| .                                                                                  | .                                                                  | .                                                                                                                                                    |
| ADDS DATA1,<br>DATA2, DATA3                                                        | (Expansion of ADDS )<br><br>ADD1 DATA1<br>ADD1 DATA2<br>ADD1 DATA3 | (Expansion of ADD1)<br>L 1, DATA1<br>A 1,=F '1'<br>ST 1, DATA1<br>L 1, DATA2<br>A 1,=F '1'<br>ST 1, DATA2<br>L 1, DATA3<br>A 1,=F '1'<br>ST 1, DATA3 |
| DATA1 DC, F '5'<br>DATA2 DC, F '10'<br>DATA3 DC, F '5'                             |                                                                    | DATA1 DC, F '5'<br>DATA2 DC, F '10'<br>DATA3 DC, F '5'                                                                                               |

Within the definition of the macro 'ADDS' are three separate calls to a previously defined macro 'ADD1'. The use of the macro 'ADD1' has shortened the length of the definition of 'ADDS' and thus had made it more easily understood. Such uses of macros result in macro expansions on multiple 'levels'.

**(2) Draw flow chart & explain working of absolute loader.**

**4M**

**Ans:**



**(Flowchart: 2 marks,  
Working: 2 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

**Working of absolute loader:** With an absolute loading scheme, the programmer and assembler perform the task of allocation, relocation and linking. Therefore, it is only necessary for the loader to read cards of the object deck and move the texts on the cards into the absolute locations specified by the assembler. There are two types of information that the object deck must communicate from the assembler to the loader.

- First, it must convey the machine instructions that the assembler has created along with the assigned core locations, (called as text cards).
- Second, it must convey the entry point of the program, which is where the loader is to transfer control when all instructions are loaded (called as transfer cards).

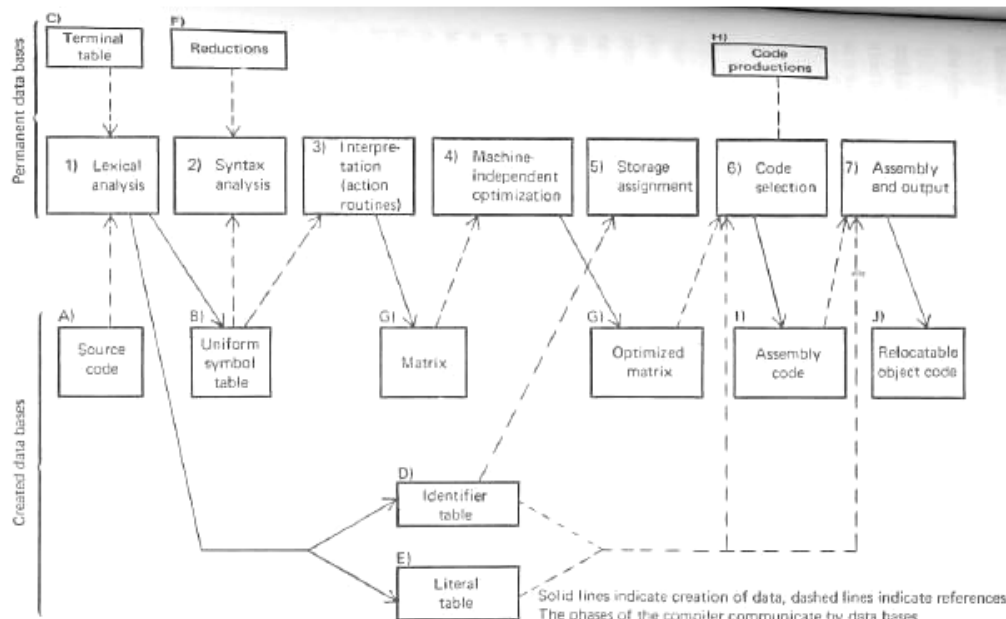
**Algorithm:**

- The object deck for this loader consists of a series of text cards terminated by the transfer cards.
- Therefore, the loader should read one card at a time, moving the text to the location specified on the card, until the transfer card is reached.
- At this point the assembled instructions are in core, and it is only necessary to transfer to the entry point specified on the transfer card.

**(3) Explain general model of a compiler.**

**4M**

**Ans:** In analyzing the compilation of simple program there are seven distinct logical problems as follows and summarized in figure below.



**(Figure: 2 marks ,Working: 2 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

|        |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |      |  |                           |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------|--|---------------------------|
|        |      | <div>1. <b>Lexical analysis</b> – Recognition of basic elements of creation of uniform symbols.</div> <div>2. <b>Syntax analysis</b> – Recognition of basic syntactic constructs through reductions.</div> <div>3. <b>Interpretation</b> – Definition of exact meaning, creation of matrix and tables by action routines.</div> <div>4. <b>Machine Independent Optimization</b> – Creation of more optimal matrix.</div> <div>5. <b>Storage Assignment</b> – Modification of identifier and literal tables. If makes entries in the matrix that allow code generation to create code that allocates dynamic storage and that also allow the assembly phase to reserve the proper amounts of STATIC storage.</div> <div>6. <b>Code Generation</b> – Use of macro processor to produce more optimal assembly code.</div> <div>7. <b>Assembly And Output</b> – Resolving symbolic addresses and generating machine language.</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |      |  |                           |
|        | (4)  | Mention four functions of storage assignment phase of compiler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 4M     |      |  |                           |
|        | Ans: | <div>The purpose of this phase is to:</div> <div>1. Assign storage to all variables referenced in the source program.</div> <div>2. Assign storage to all temporary locations that are necessary for intermediate result, e.g the results of matrix lines. These storage references were reserved by the interpretation phase and did not appear in the source code.</div> <div>3. Assign storage to literals.</div> <div>4. Ensure that the storage is allocated and appropriate locations are initialized (Literals and any variables with the initial attribute)</div> <div>The storage allocation phase first scans through the identifier table, assigning locations to the storage allocation phase first scans through the identifier table, assigning locations to each entry with a storage class of static. It uses a location counter, initialized at zero, to keep track of how much storage it has assigned.</div> <div>Whenever it finds a static variable in the scan, the storage allocation phase does the following four steps:</div> <div>1. Updates the location counter with any necessary boundary alignment.</div> <div>2. Assigns the current value of the location counter to the address field of the variable.</div> <div>3. Calculate the length of the storage needed by the variable (by examining its attributes).</div> <div>4. Updates the location counter by adding this length to it. Once it has assigned relative address to all identifiers requiring STATIC storage locations, this phase creates a matrix entry:</div> <div><table><tr><td>STATIC</td><td>Size</td><td></td></tr></table></div> <div>5. This allows code generation to generate the proper amount of storage. For each</div> | STATIC | Size |  | (Explanation:<br>4 marks) |
| STATIC | Size |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |      |  |                           |



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

variable that requires initialization, the storage allocation phase generates a matrix entry:

|            |          |  |
|------------|----------|--|
| Initialize | Variable |  |
|------------|----------|--|

6. This tells code generation to put into the proper storage location the initial values that the action routines saved in the identifier table. A similar scan of the identifier table is made for automatic storage and controlled storage. The scan enters relative location for each entry. An “automatic” entry and a “controlled” entry are also made in the matrix. Code generation use the relative location entry to generate the address part of instructions. No storage is generated at compile time for automatic or controlled. However, the matrix entry automatic does cause code to be generated that allocates this storage at execution time, i.e., when the generated code is executed, it allocates automatic storage.

|     |      |  |
|-----|------|--|
| LIT | Size |  |
|-----|------|--|

7. The literal table is similarly scanned and location are assigned to each literal, and a matrix entry is made. Code generation generates storage for all literals in the static area and initializes the storage with the values of the literals. Temporary storage is handled differently since each source statement may reuse the temporary storage (intermediate matrix result area) of the previous source statement. A computation is made of the temporary storage that is required for each source statement. The statement required the greatest amount of temporary storage determines the amount that will be required for the entire program. A matrix entry is made of the form this enables the code generation phase to generate code to create the proper amount of storage.

|           |      |  |
|-----------|------|--|
| AUTOMATIC | Size |  |
|-----------|------|--|

8. Temporary storage is automatic since it is only referenced by the source program and only needed while the source program is active.

**(5) Explain single pass algorithm for macro processing.**

**4M**

**Ans:** If we wanted to provide for macro definitions within macros. The basic problem here is that inner macro is defined only after the outer one has been called in order to provide for any use of the inner macro we would have to repeat both the macro-definition and the macro-call passes. However there is a simpler solution that has added advantages of reducing all macro processing to a single pass. There are two additional variables introduced in the one –pass design a macro definition input (MDI) indicator and a

**(Explanation:  
4 marks )**



Macro Definition Level Counter (MDLC). The MDI and MDLC are switches (counters) used to keep track of macro calls and macro definitions. The MDI indicator has the value “ON” during expansion of a macro call and the value “OFF” at all other times. The actual expansion of macro calls is performed in the read box. READ tests the switch MDI. If it is “ON”, lines are read from the Macro Definition Table (MDT). The reading of a MEND line indicates the end of a macro and terminates expansion of call: MDI is reset to “OFF” and the next line is obtained from the regular input stream. Note that lines returned by READ may include macro definition’s; expanded macro code comes out of READ looking just like any other code and may therefore include macro definitions. The macro definition level counter is incremented by 1 when a MACRO pseudo-op is encountered and decremented by 1 when a MACRO pseudo-op occurs. The MDLC is used to insure that the entire macro definition, including MACROs and MENDs, gets stored in MDT.

**Working of Macro:**

1. Start
2. Initialize MDTC = 1  
MNTC = 1  
MDI = OFF  
MDLC = 0
3. Read- perform read operation of macro.
4. Search MNT for match found of operation code.
5. Is macro name found ? macro call  
If yes MDI = ‘ON’  
MDTP = MDT Index from MNT entry  
Setup ALA  
Goto step 3  
If no go to step 6
6. Is macro pseudo-op ? macro definition,  
If yes MDLC = MDLC + 1  
If no then read code again.
7. Store macro name and current value of MDTC in MNT entry number in MNTC.
8. Increment MNTC by 1  
Prepare ALA
9. Store macro name card in MDT  
Increment MDTC by 1  
Increment MDLC by 1
10. Read - perform read operation of macro.
11. Substitute index notation for arguments in definition.  
Enter line is MDT  
Increment MDTC by 1
12. Is MACRO pseudo-op  
If yes increment MDLC by 1 and goto step 10.  
If no  
Is MEND pseudo-op ?  
If yes



|  |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                  |
|--|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
|  |             | <p>Decrement MDLC by 1<br/>Goto step 13.<br/>If no, goto step 10</p> <p><b>13.</b> If MDLC = 0<br/>If yes, goto step 3<br/>If no, goto step 10</p> <p><b>14.</b> Write expanded code in source file card.</p> <p><b>15.</b> Is END pseudo-op ?<br/>If yes<br/>Supply expanded source file to assembler processing<br/><br/>If no, goto step 3.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                  |
|  | <b>(6)</b>  | <b>Define parser. Draw the parse tree for string 'abccd' using top-down parser.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <b>4M</b>                                        |
|  | <b>Ans:</b> | <p><b>Parser:</b> A parser is a program that receives input in the form of sequential source program instructions, interactive online commands, markup tags, etc and breaks them up into parts such as mnemonics, symbols, objects, methods, etc that can then be managed by other phases of compiler. Parser is also called as "Syntax analyzer".</p> <p><b>Parse tree for the string 'abccd' using top down parser.</b><br/>String is "abccd"<br/><b>Assume:</b><br/> <math>S \rightarrow xyz \mid aBC</math><br/> <math>B \rightarrow b \mid bc</math><br/> <math>C \rightarrow dc \mid cd</math></p> <p><b>Steps</b></p> <ul style="list-style-type: none"> <li>• Assertion 1 : abccd matches S <ul style="list-style-type: none"> <li>• Assertion 2: abccd matches xyz: <ul style="list-style-type: none"> <li>▪ <b>Assertion is false. Try another.</b></li> </ul> </li> <li>• Assertion 2 : abccd matches aBC i.e bccd matches BC:</li> <li>• Assertion 3 : bccd matches cC i.e cccd matches C: <ul style="list-style-type: none"> <li>▪ Assertion 4 : ccd matches dc: <ul style="list-style-type: none"> <li>• False.</li> </ul> </li> <li>▪ Assertion 4 : ccd matches dc: <ul style="list-style-type: none"> <li>• False.</li> </ul> </li> </ul> </li> <li>• Assertion 3 is false. Try another.</li> <li>• Assertion 3 : ccd matches bcC i.e cd matches C:</li> <li>• Assertion 4 : cd matches dc: <ul style="list-style-type: none"> <li>▪ False.</li> </ul> </li> <li>• Assertion 4 : cd matches cd:</li> <li>• Assertion 4 is true.</li> </ul> </li> </ul> | <b>(Definition: 1 mark, Parse Tree: 3 marks)</b> |



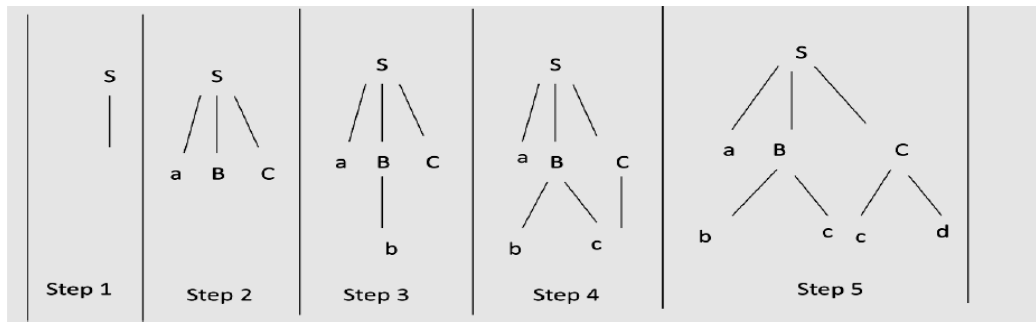
**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

- Assertion 3 is true.
- Assertion 2 is true.
- Assertion 1 is true.



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| 6.   | Solve any FOUR:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 16Marks                                                                                              |
| (1)  | Explain assembly phase of a compiler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 4M                                                                                                   |
| Ans: | <ul style="list-style-type: none"><li>• After code generation phase, next phase is assembly phase. The task of assembly phase depends on how much has been done in code generation.</li><li>• If a lot of work has been done in code generation, then the assembly phase must resolve labels references in object program, format the object deck, and format the appropriate information for the loader.</li><li>• If code generation has simply generated symbolic machine instructions and labels, the assembly phase must (1) resolve label references, (2) calculate addresses, (3) generate binary machine instructions, and (4) generate storage, convert literals.</li></ul> <p>Databases:</p> <ol style="list-style-type: none"><li>1. Identifier table: assembly phase uses this database to enter the value of all labels into identifier table.</li><li>2. Literal table: places the literal on appropriate TXT cards</li><li>3. Object code: the output of code generation.</li></ol> <p>Algorithm:</p> <ol style="list-style-type: none"><li>1. A simple assembly phase scans the object code, resolving all label references and producing the TXT cards.</li><li>2. It then scans the identifier table to create the ESD cards.</li><li>3. The RLD cards are created using the object code, the ESD cards and the identifier table</li></ol> | <p>(2 marks<br/>Explanation:<br/>2 marks<br/>Databases: , 1<br/>mark,<br/>Algorithm: 1<br/>mark)</p> |

**MODEL ANSWER****SUMMER- 17 EXAMINATION****Subject Title: System Programming**

Subject Code:

**17634**

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                        |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| (2)  | Sort following numbers in ascending order by bucket sort:<br>78,354,51,278,63,89,312,12                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 4M                     |
| Ans: | <div style="text-align: center;"> <p>0-100                      101-200                      201-300</p> <p>Sort individual buckets</p> <p><b>Sorted list:</b> 12, 51, 63, 78,89, 278, 3</p> </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | (4 marks )             |
| (3)  | Explain features of macro facility.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 4M                     |
| Ans: | <p><b>Following are features of a MACRO facility:</b></p> <ol style="list-style-type: none"> <li>1. Macro instruction Arguments.</li> <li>2. Conditional Macro Expansion</li> <li>3. Macro calls within Macros</li> <li>4. Macro instruction Defining Macros</li> </ol> <p><b>1. Macro Instruction Arguments:</b> The macro facility presented is capable of inserting block of instructions in place of macro calls. All of the calls to any given macro will be replace by identical blocks. This lacks flexibility: there is no way for a specific macro call to modify the coding that replaces it. An important extension of this facility consists of providing for arguments. An important extension of this facility consists of providing for arguments or parameters in macro calls. Corresponding macro dummy arguments will appear in macro definitions.</p> <pre> . . . A      1,DATA1 A      2,DATA1 A      3,DATA1 . . . A      1,DATA2 A      2,DATA2 A      3,DATA2 </pre> | (Each feature: 1 mark) |

**MODEL ANSWER****SUMMER- 17 EXAMINATION****Subject Title: System Programming**

Subject Code:

**17634**

```

.
.
.
DATA    F"5"
1 DC
DATA    F"10"
2 DC

Macro INCR has One
Argument

MACRO
INCR      &ARG
A        1,&ARG
A        2,&ARG
A        3,&ARG
MEND

.
.
.
INCR      DATA1      Use DATA1 as
                        operand

.
.
.
INCR      DATA2      Use DATA2 as
                        operand

.
.
.
DATA1 DC    F"5"
DATA2 DC    F"10"
.
.
.

```

In this case the instruction sequences are very similar but not identical. The first sequences performs an operation using DATA1 as operand; the second using DATA2. They can be considered to perform the same operation with a variable parameter, or argument. Such parameter is called a macro instruction argument or dummy arguments. It is specified on the macro name line and distinguished by the ampersand which is always its first character.

**2. Conditional Macro Expansion:** Two important macro processor pseudo-ops, AIF and AGO, permit conditional reordering of the sequence of macro expansion. This allows conditional selection of the machine instructions that appear in expansions of macro call. AIF is conditional branch pseudo-o; it performs an arithmetic test and



branches only if the tested condition is true. The AGO is an unconditional branch pseudo ops or 'go to' statement. It specifies a label appearing on some other statement in the macro instruction definition; the macro processor continues sequential processing of instruction with the indicated statement. These statements are directives to the macro processor and do not appear in macro expansion.

**3. Macro call within Macros:** Since macro calls are "abbreviations" of instruction sequences, it seems reasonable that such abbreviations should be available within other macro definitions. Macro calls within macros can involve several levels. In fact, conditional macro facilities make it possible for a macro to call itself. So long as this does not cause an infinite loop so long as at some point the macro having been called for the nth time, decides not to call itself again perfectly.

**4. Macro Instructions Defining Macros** Macros are generalized abbreviations for instruction sequences, nothing that it seems reasonable to permit any valid statements in the abbreviated sequence, including macro definitions. In this manner a single macro instruction might be used to simplify the process of defining a group of similar macros.

**(4) List the steps for binary search algorithm. List the best, worst and average case complexity.**

**4M**

**Ans:** **Binary Search Algorithm:** A more systematic way of searching an ordered table. This technique uses following steps for searching a keywords from the table.

1. Find the middle entry ( $N/2$  or  $(N+1)/2$ )
2. Start at the middle of the table and compare the middle entry with the keyword to be searched.
3. The keyword may be equal to, greater than or smaller than the item checked.
4. The next action taken for each of these outcomes is as follows  
If equal, the symbol is found  
If greater, use the top half of the given table as a new table search  
If smaller, use the bottom half of the table.

**Example:**

The given nos are: 1,3,7,11,15

To search number 11 Indexing the numbers from list [0] upto list[5]

**Pass 1**

Low=0, High = 5, Mid=  $(0+5)/2 = 2$

So

**(Algorithm: 2 marks  
Complexity: 2 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

list[2] = 3 is less than 7

**Pass 2**

Low= (Mid+1)/2 i.e (2+1)/2 = 1, High = 5, Mid= (1+5)/2 = 6/2 = 3

So list [3] = 11

and the number is found.

**List the best, worst and average case complexity:**

- Binary Search can be analyzed with the best, worst, and average case number of comparisons. These analyses are dependent upon the length of the array, so let  $N = |A|$  denote the length of the Array A.
- The numbers of comparisons for the recursive and iterative versions of Binary Search are the same, if comparison counting is relaxed slightly. For Recursive Binary Search, count each pass through the if-then-else block as one comparison. For Iterative Binary Search, count each pass through the while block as one comparison.

- 1. Best case - O (1) comparisons** In the best case, the item X is the middle in the array A. A constant number of comparisons (actually just 1) are required.
- 2. Worst case - O (log n) comparisons** In the worst case, the item X does not exist in the array A at all. Through each recursion or iteration of Binary Search, the size of the admissible range is halved. This halving can be done ceiling (lg n ) times. Thus, ceiling (lg n ) comparisons are required.
- 3. Average case - O (log n) comparisons** To find the average case, take the sum over all elements of the product of number of comparisons required to find each element and the probability of searching for that element. To simplify the analysis, assume that no item which is not in A will be searched for, and that the probabilities of searching for each element are uniform.

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                               |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| (5)         | <b>Describe the term token with respect to Lexical Analysis.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>4M</b>                     |
| <b>Ans:</b> | <ul style="list-style-type: none"> <li>• The first phase of compiler is lexical analysis. It works as a text scanner. This phase scans the source code as a stream of characters and converts it into meaningful lexemes. Lexical analyser represents these lexemes in the form of tokens as:<br/>&lt;token-name, attribute-value&gt;</li> <li>• Algorithm of Lexical Analysis phase of compiler is as follows <ul style="list-style-type: none"> <li>○ The first tasks of the lexical analysis algorithm are to the input character string into token.</li> <li>○ The second is to make the appropriate entries in the tables.</li> <li>○ A token is a substring of the input string that represents a basic element of the language. It may contain only simple characters and may not include another token. To the rest of the compiler, the token is the smallest unit of currency. Only lexical analysis and the output processor of the assembly phase concern themselves with such elements as characters. Uniform symbols are the terminal symbols for syntax analysis.</li> </ul> </li> </ul> | <b>(Explanation: 4 marks)</b> |

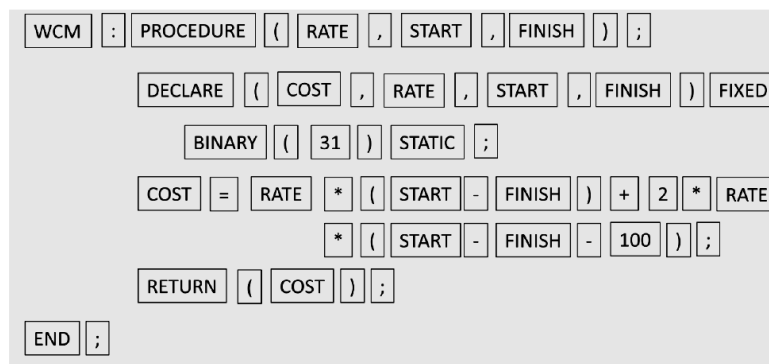


**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code: **17634**

- Lexical analysis recognizes three types of token: Terminal symbols, possible identifiers, and literals.
- It checks all tokens by first comparing them with the entries in the terminal table. Once a match is found, the token is classified as a terminal symbol and lexical analysis creates a uniform symbol of type „TRM., and inserts it in the uniform symbol table. If a token is not a terminal symbol, lexical analysis proceeds to classify it as a possible identifier or literal. Those tokens that satisfy the lexical rules for forming identifiers are classified as “possible identifiers”.
- **Example:**  
**Consider following program**  
WCM: PROCEDURE(RATE,START,FINISH);  
DECLARE (COST,RATE,START,FINISH) FIXED BINARY (31)STATIC;  
COST = RATE \* (START-FINISH) + 2\*RATE\*(START-FINISH-100);  
RETURN (COST);  
END;



**(6) Explain Conditional Macro expansion.**

**4M**

- Ans:**
- Two important macro-processor pseudo-ops AIF and AGO permit conditional reordering of the sequence of macro expansion. This allows conditional selection of the machine instructions that appear in expansions of Macro call. Consider the following program.

Loop 1 A1, DATA 1  
A2, DATA 2  
A3, DATA 3  
.

Loop 2 A1, DATA 3  
A2, DATA 2  
.

Loop 3 A1, DATA1  
.

**(Explanation:  
4 marks)**



**MODEL ANSWER**  
**SUMMER- 17 EXAMINATION**

**Subject Title: System Programming**

Subject Code:

**17634**

DATA 1 DC F'5'

DATA 2 DC F'10'

DATA 3 DC F'15'

- In the below example, the operands, labels and the number of instructions generated change in each sequence. The program can be written as follows:-

```
.  
.   
.   
MACRO  
&ARG0 VARY &COUNT,&ARG1,&ARG2,&ARG3  
&ARG0 A 1,&ARG1  
AIF (&COUNT EQ 1).FINI  
A 2,&ARG2  
AIF (&COUNT EQ 2).FINI  
A 3,&ARG3  
.FINI MEND EXPANDED SOURCE  
..  
..  
..  
LOOP1 VARY 3,DATA1,DATA2,DATA3 LOOP1 A 1,DATA1  
. A 2,DATA2  
. A 3,DATA3  
..  
LOOP2 VARY 2,DATA3,DATA2 LOOP2 A 1,DATA3  
. A 2,DATA2  
..  
..  
LOOP3 VARY 1,DATA1 LOOP3 A 1,DATA1  
.  
.  
DATA1 DC F'5'  
DATA2 DC F'10'  
DATA3 DC F'15'
```

- Labels starting with a period (.) such as .FINI are macro labels and do not appear in the output of the macro processor.
- The statement AIF (& COUNT EQ1) .FINI directs the macro processor to skip to the statement labelled .FINI if the parameter corresponding to & COUNT is a1; otherwise the macro processor is to continue with the statement following the AIF pseudo-ops. AIF is a conditional branch pseudo-op; it performs an arithmetic test and branches only if the tested condition is true.
- AGO is an unconditional branch pseudo-op or 'Go to' statement. It specifies a label appearing on some other statement. AIF & AGO controls the sequence in which the macro processor expands the statements in macro instructions.