Chapter 2 DATA STRUCTURES

Scope of the Syllabus — **Probable marks : 17**

- Introduction to data structures.
- Data structure operations.
- Algorithmic notations.
- Control structures.
- Arrays-Representation in memory, Traversing, Deleting, Sorting, Binary search in an array, Pointer arrays, Records in memory using array.
- Linked list - Representation in memory
- Trees, Binary trees - Representing binary tree in memory.

INTRODUCTION TO DATA STRUCTURES

Q. 1 Define the following terms :

(i) Data :

Ans. Data are simply values or set of values.

(ii) Group items : (March 2018)

Ans. Data items which are divided into subitems are called as group items. e.g. Date may be divided into three subitems - day, month and year. So Date becomes group item.

(iii) Elementary items : (March 2018)

Ans. Data items which are not divided into subitems are called as elementary items. e.g. pincode number cannot be divided into subitems. So it is elementary item.

(iv) Entity : (March 2018)

Ans. An entity is something that has certain attributes or properties which may be assigned values.

The values themselves may be numeric or nonnumeric.

e.g. A Bio-data sheet mainly contains :

Attributes ⟶	Name	Age	Sex	Education
Values ⟶	Atul	22	M	B.E. (Computer)

(v) Field : (March 2005, 2009,2016; Oct. 2007,2014; July 201

Field is a single elementary unit of information representing an attribute of an entity.

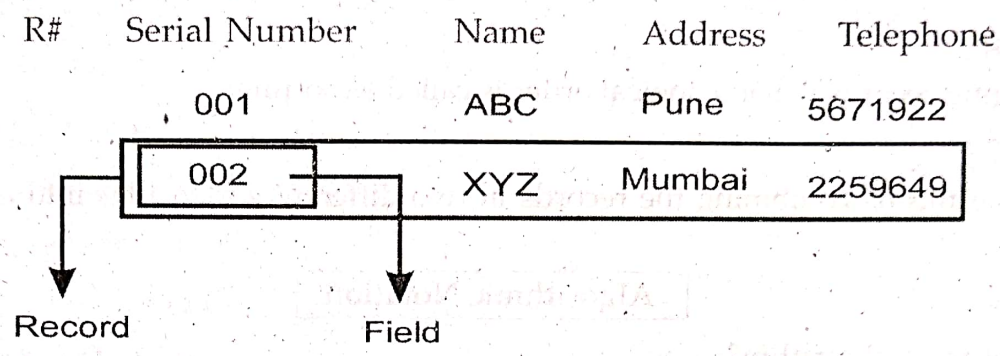
(vi) Record : (March 2005, 2009,2016; Oct. 2007,2014; July 2019

Record is a collection of field values of a given entity.

(vii) File : (March 2005, 2009,2016; Oct. 2007,2014 ; July 2019

File is the collection of records of the entities in a given entity set.

e.g.



Q. 2 What is a data structure ? (Mar. 2006,2015; Oct. 2002, 2004, March 2018, Dec. 2020

Ans. :

i) Data may be organized in many different ways. Data structure is the way in which different data elements are logically related.

ii) Collection of data elements forming an organisation characterized by the accessing functions is called data structure.

iii) The data structure should be simple and it should show the relationship between data elements.

iv) Types :

(i) Linear data structure (ii) Non-linear data structure.

In linear data structure, data elements are stored in consecutive memory locations or by using linked representation. e.g. arrays, linked list.

In non-linear data structures the linear order cannot be maintained between data elements. Generally data elements have hierarchical relationship between them. e.g. trees

v) Computer language provides different data structures like arrays, stack, queue, tree etc.

Data Structure Operations

Q. 3 Explain in brief any six data structure operations.

(Oct. 2002,04,06,12,15; Mar. 2002,06,12, July 2017, 19, March 2020, Dec. 2020

Ans. : The data appearing in data structures are processed by means of certain operations like :

(i) Traversing :

Accessing each record or element exactly once, so that it can be processed is called as traversing.

For e.g. multiplying each element of an array by 6.

(ii) Inserting :

Adding a new record to the existing structure is called as inserting.

(iii) **Deleting :**

Removing a record from the existing structure is called as deleting.

(iv) **Searching :**

Finding the location of a record with given key values or finding the locations of all records which satisfy one or more conditions is called as searching.

(v) Sorting :

Arranging records in some logical order is called as sorting.

(vi) Merging :

Merging means combining the records in two different sorted files into a single sorted file.

Algorithmic Notation

Q. 4 What is an algorithm?

Ans. : **(July 2017)**

i) An algorithm is a finite step by step list of well-defined instructions for solving a particular problem.

ii) An algorithm consists of two parts :

(a) First part is a paragraph which tells the purpose of algorithm. In this part, we define variables in algorithm and lists the input data.

(b) The second part of algorithm consists of steps in algorithm that are executed one after the another, generally beginning with step 1, unless stated otherwise. The control can be transferred to step n, by the statement "go to step n".

The algorithm is completed, when the statement 'Exit' or 'Stop' is encountered.

iii) e.g. Algorithm to find largest element in array.

Largest [DATA, N, MAX]

Here, DATA is a linear array with N elements. This algorithm finds the largest element MAX of DATA.

Step 1 : [Initialize counter]

set k: = 1 and Max: = DATA [1]

Step 2 : [Compare and Update]

If MAX < DATA [k + 1],

then : MAX: = DATA [k + 1]

[End of If structure]

Step 3 : [Increment counter] set k: = k + 1

Step 4 : [Test counter] If k < N, then : go to step 2

[End of If structure]

Step 5 : Write : MAX

Step 6 : Exit

Q. 5 Write an algorithm to find smallest element in an array. **(March 2015, 3)**

Ans. :

Algorithm to find Smallest element in an array

Smallest[DATA , N, MIN]

Here DATA is a linear array with N elements. This algorithm finds smallest element NIN of DATA

Step 1 : [Initialize counter]

set k = 1 and MIN = DATA[1]

Step 2 : [Compare and Update]

If MIN > DATA[k +1]

then MIN = DATA[k +1]

[End of If Statement]

Step 3 : Set k = k+1

Step 4 : If k < N then go to step 2

[End of If Statement]

Step 5 : Write MIN

Step 6 : Exit

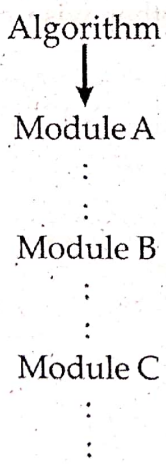
Q. 6 Describe sequence logic or sequential flow.

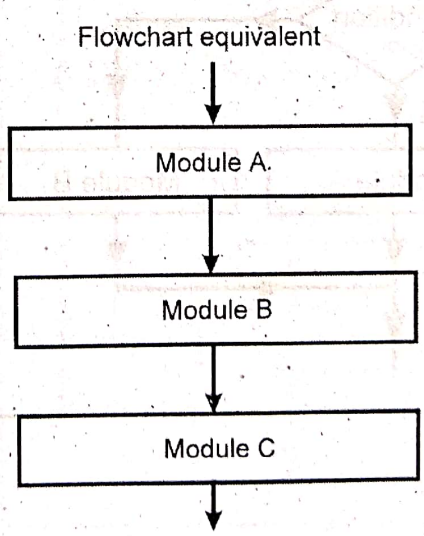
Ans. :

i) In the sequence logic, modules are executed sequentially, one after the another.

ii) The sequence may be present explicitly by means of numbered step or by the order in which modules are written.

iii) In short in sequential logic or sequential flow, modules of an algorithm are executed one after the another.

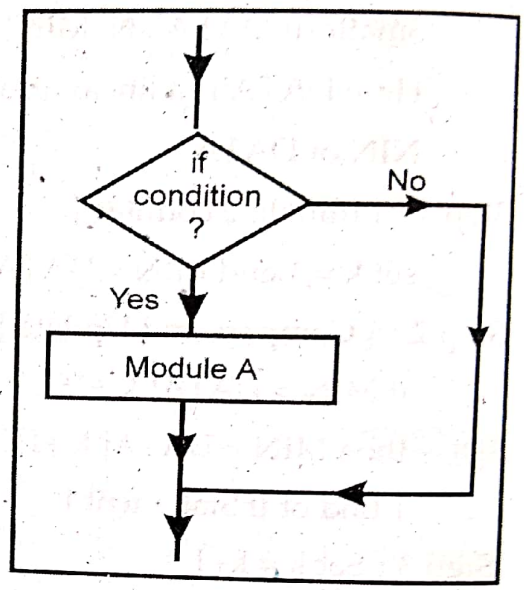




Q. 7 Describe conditional flow or selection logic. **(July 2019)**

Ans. : Selection logic uses number of conditions, which cause selection of one out of several alternative modules. The structure which implement this type of logic is known as selection logic, or conditional structure.

There are three types of conditional structures :



(i) Single alternative :

This has the form :

```
If condition, then :
    [module A]
[End of If structure]
```

The logic of this structure is as follows :

If condition is satisfied (true) then module A, which consists of number of statements, is executed.

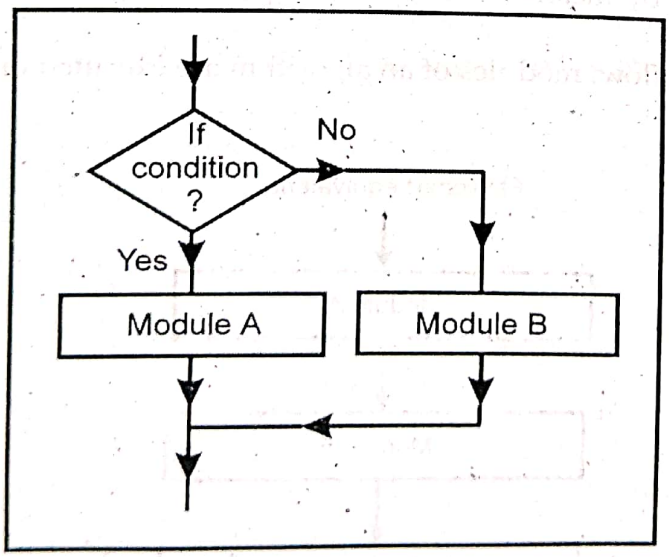
Otherwise, module A is skipped and next module of algorithm is executed.

(ii) Double alternative :

This structure has the form :

```
If condition, then :
    [module A]
Else :
    [module B]
[End of If structure]
```

The logic of this structure is as follows



If the condition is satisfied, then module A will be executed other wise module B will be executed.

(iii) Multiple alternative :

This structure has the form :

```
If condition (1), then :
    [module A1]
else if condition (2), then :
    [module A2]
        :
        :
else if condition (n), then :
    [module An]
else :
    [module B]
[End of If structure]
```

The logic of this structure allows only one module to be executed. The module following the condition, which is satisfied the condition will be executed. If no condition is satisfied, then the module, which follows last Else statement will be executed.

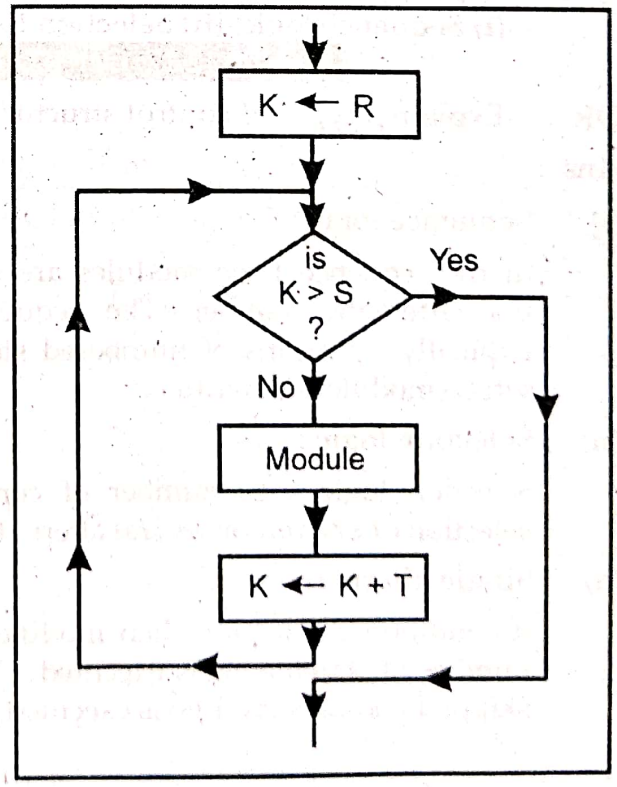
Q. 8 Describe logic of Repeat-For loop. **(July 2019)**

Ans. : The repeat-for loop has the form :

```
Repeat for K = R To S by T :
    [module]
[End of For loop]
```

Here, K is called index variable, R and S are initial and final values of K and T is increment. The logic of this structure is as follows :

At first, the body of loop i.e. module will be executed with K = R and then with K = R + T, then with K = R + 2T and so on, until K < = S. The loop ends when K > S. If T is negative then K decreases in value and loop ends when K < S.

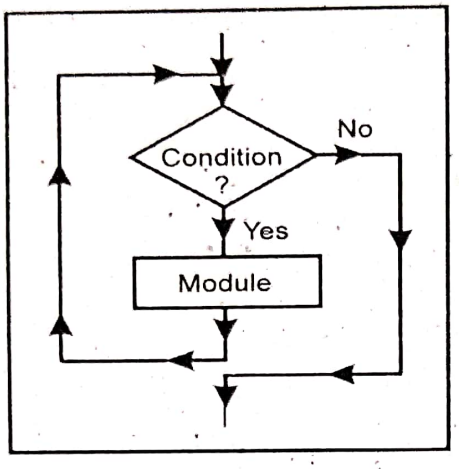


Q. 9 Explain Repeat-While structure.

Ans. : The repeat-while loop has the form :

```
Repeat While condition :
    [module]
[End of loop]
```

Here, body of loop i.e. module is executed repeatedly, unit the condition is satisfied.



There must be a statement before the structure that initializes the condition controlling the loop and there must be a statement in the body of the loop that changes the condition.

For e.g. Find largest element in array.

Given a nonempty array DATA with N numerical values. This algorithm finds the location LOC and the value MAX of the largest element of DATA.

1. Set K := 1, LOC := 1, MAX : = DATA[1]
2. Repeat step 3 and 4 while K < = N :
3. If MAX < DATA [K], then :
   set LOC := K and
   set MAX := DATA[K]
   [End of If structure]
4. Set K := K + 1
   [End of step 2 loop]
5. Write : LOC, MAX
6. Exit

Q. 10 Explain with flowcharts the following control structures :
(i) Sequence logic, (ii) Selection logic, (iii)Iteration logic
(Mar. 09,12,17; Oct. 03, 04, 05,11,14,21; July 18, 19

OR Explain 3 types of control structures used for flow of control

Ans. :

(i) Sequence logic :

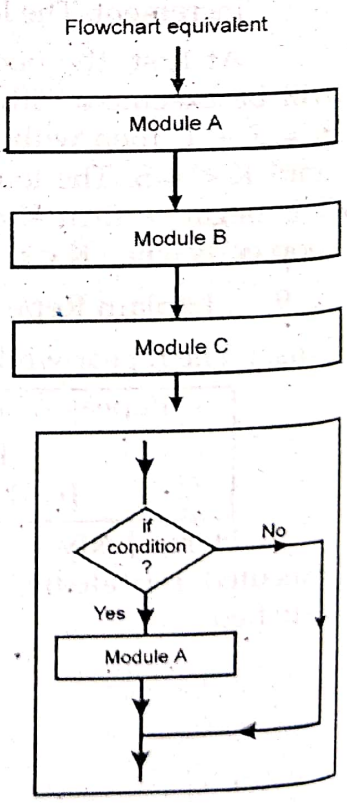
In the sequence logic modules are executed sequentially, one after the another. The sequence may be present explicitly by means of numbered step or by the order in which modules are written.

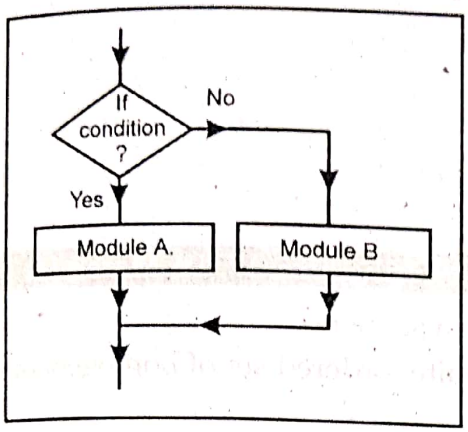
(ii) Selection logic :

Selection logic uses number of conditions, which cause selection of one out of several alternative modules.

(a) Single alternative :

If condition is satisfied then module A, which consists of number of statements, is executed. Otherwise module A is skipped and next module is executed.



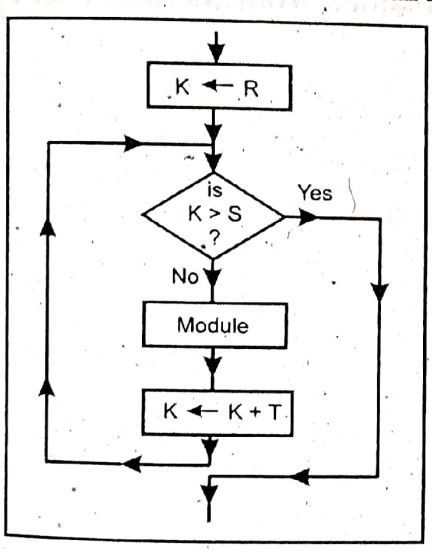


(b) Double alternative :

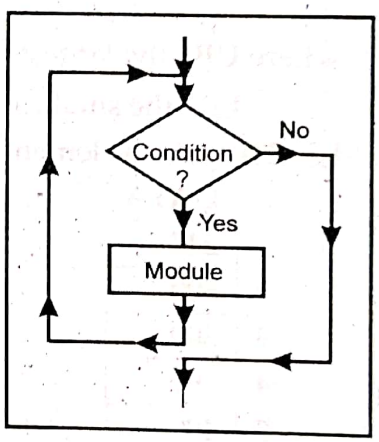
If the condition is satisfied, then module A will be executed otherwise module B will be executed.

(c) Iteration logic :

Here certain module is executed repeatedly until condition satisfies. At first, the body of loop i.e. module will be executed with K = R and then with K = R + T, then with K = R + 2T and so on until K = S. The loop ends when K > S.



The repeat-while structure has the form :



Here module is executed until the condition is satisfied.

Q. 11 Write an algorithm to find solutions of quadratic equation $Ax^2 + Bx + C = 0$ where $A \neq 0$ (Oct. 2002, July 2017)

Ans. : The algorithm inputs the coefficients A, B, C of a quadratic equation and outputs the real solution, if any.

1. Read A, B, C
2. Set $D := B^2 - 4 * A * C$
3. If $D > 0$, then :
   a) set $X1 := -B + \sqrt{D} / 2A$
      and $X2 := -B - \sqrt{D} / 2A$
   b) Write : X1, X2

Else if $D = 0$, then :

   a) set $X := -B/2A$
   b) Write : 'UNIQUE SOLUTION' , X

Else

Write : 'NO REAL SOLUTION'

[End of If structure]

4. Exit

ARRAY

Q. 12 What are linear arrays ? (Oct. 2006,13,14; Mar.2015, July 2016, 17, 19, March 18, 19

Ans. : A data structure is said to be linear if its elements form a sequence.

A linear array is the data structure which consists of finite, ordered set of homogeneous data elements such that :

1. The elements of the array are referenced respectively by an index set (subscript) consisting of 'n' consecutive numbers.
2. The elements of the array are stored respectively in successive memory locations.
3. The number 'n' of the elements is called length or size of array.

In general, the size or length of the array can be obtained from the index set by the formula :

$$\text{Length} = UB - LB + 1$$

where UB - the largest index called Upper Bound.

LB - the smallest index called Lower Bound.

e.g. Let DATA be 5 element linear array as follows :

DATA

1	247
2	500
3	600
4	399
5	499

The element of an array may be denoted by the subscript notation such that :

DATA [3] = 600

In C++, array is declared as -

int data [100]; which specify an array data of 100 integers.

Q. 13 How arrays are represented in memory ? (Oct. 2014)

Ans. :

i) The elements of linear array are stored in consecutive memory locations.

ii) Computer does not need to keep track of the address of every element of array. It just requires the address of first element of array, LA, denoted by Base (LA) and called the base address of linear array LA.

iii) Using this base address, the computer calculates address of any element of array by using the formula.

$$LOC(LA[K]) = Base(LA) + W(K - LB)$$

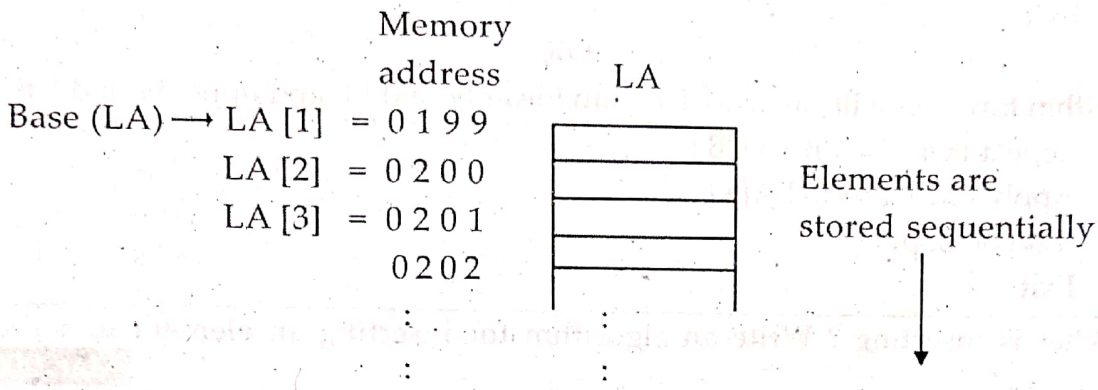
where,

LOC (LA[K]) is address of K^{th} element of LA

W is number of words per memory location for LA

and LB is lower bound i.e. smallest index of LA.

iv) The memory representation of an array is shown in figure below :-



Q. 14 Consider the array AUTO, which records number of automobiles from 1932 through 1984. Suppose Base (AUTO) = 200 and W = 4 words. Then,

LOC (AUTO [1932]) = 200, LOC (AUTO [1933]) = 204

LOC (AUTO [1934]) = 208

Calculate address at which 1965's record is stored.

Ans. :

Given : - K = 1965

Base address = 200

W = 4

LB = 1932

The address of the array element for the year 1965 can be obtained –

LOC (AUTO [1965]) = Base (AUTO) + W (1965 – LB)

= 200 + 4 (1965 – 1932) = 332

Q. 15 What is traversing an array ? Give the algorithm for traversing a linear array.

(Oct. 2005,12,15; March 2006, March 2018; July 18

Ans. :

Traversing an array means accessing with each element of array only at once, so that it can be processed.

Algorithm : Traversing a linear array.

Here LA is a linear array with lower bound LB and upper bound UB. Following algorithm apply operation PROCESS to each element of LA.

Step 1 : [Initialize counter]

set K : = LB

Step 2 : Repeat steps 3 and 4 while K ≤ UB :

Step 3 : [Visit element]
Apply PROCESS to LA[K]
Step 4 : [Increment counter]
set K := K + 1
[End of step 2 loop]
Step 5 : Exit

OR

This algorithm traverses a linear array LA with lower bound LB and upper bound UB.

Step 1 : Repeat FOR K = LB To UB :
Apply PROCESS to LA[K]
[End of loop]
Step 2 : Exit

Q. 16 What is inserting ? Write an algorithm for inserting an element to a linear array. (Mar. 2009,11,16)

Ans. : (Oct.2015)

i) Inserting refers to the operation of adding an element to the existing elements of array.

ii) The element can be easily inserted at the end of array. But for insertion in the middle of array, it is required to move the elements of array one byte forward.

iii) The following algorithm inserts a data element ITEM into the Kth position in an array LA with N elements.

Algorithm :

INSERT (LA, N, K, ITEM)

Here LA is a linear array with N elements and K is a positive integer, such that K δ N. This algorithm inserts an element ITEM at K^{th} position in LA.

Step 1 : [Initialize counter]
Set J := N
Step 2 : Repeat steps 3 and 4 while J ≥ K :
Step 3 : [Move J^{th} element forward]
Set LA[J + 1]: = LA[J]
Step 4 : [Decrement counter]
Set J := J – 1
[End of step 2 loop]
Step 5 : [Insert the element]
Set LA[K] := ITEM
Step 6 : [Reset N]
Set N := N + 1
Step 7 : Exit

Q. 17 What is deleting ? Write an algorithm for deletion of an element from an array. (Oct. 07)

Ans. :

i) Deleting means removing an element from the existing elements of an array.

ii) Deletion at the end of an array is easier. But, if to delete an element from mid of array, then to move the elements of array one location upward.

iii) Algorithm : DELETE (LA, N, K, ITEM)

Here LA is a linear array with N elements and K is a positive integer, such that K = N. This algorithm deletes K^{th} element from LA and assigns it to variable ITEM.

Step 1 : Set ITEM := LA[K]

Step 2 : Repeat For J = K to N – 1 :

[Move $(J + 1)^{st}$ element backward]

Set LA[J] := LA[J + 1]

[End of loop]

Step 3 : [Reset number N of elements in LA]

Set N := N – 1

Step 4 : Exit

Q. 18 Suppose a company keeps a linear array year, such that year [K] contains number of employees in year K. Write a module for each of the following tasks :

(a) To print each of the years in which no employee was born.

(b) To find the number N of years in which no employee was born.

(c) To find the number NL of employees, who will be atleast L years old at the end of year 1984.

Linear array year contain elements from 1920 to 1970.

Ans. :

(a) To print each of the years in which no employee was born.

1. Repeat for K := 1920 to 1970 :

if year [K] = 0, then :

write : K

[End of If structure]

[End of loop]

2. Exit

(b) To find number N of years, in which no employee was born.

1. Set N := 0

2. Repeat for K = 1920 To 1970 :

If year [K] = 0, then :

N := N + 1

[End of If structure]

[End of loop]

3. Write : N

4. Exit

(c) To find number of employees NL, who will be at least L years old at the end of year 1984 we want the number of employees born in year 1984-L or earlier.

1. Set NL := 0
2. Set X := 1984 - L
3. Repeat For K = 1920 To X :
   Set NL := NL + year [K]
   [End of loop]
4. Write : NL
5. Exit

Q. 19 Explain Bubble sort algorithm with suitable example.

(March 2002, 05, 08, 12, 17, 20; Oct. 2005, 2008

Ans. : Algorithm :

Bubble Sort (DATA, N)

Here DATA is a linear array with N elements. This algorithm sorts elements of DATA in ascending order.

Step 1 : Repeat steps 2 and 3 for K := 1 To N - 1 :

Step 2 : Set Ptr := 1

Step 3 : Repeat While Ptr ≤ N - K :

(a) If DATA [Ptr] > DATA [Ptr + 1], then interchange

DATA [Ptr] and DATA [Ptr + 1]

[End of If structure]

(b) [increment pointer]

Set ptr := ptr + 1

[End of inner loop]

[End of outer loop]

Step 4 : Exit

Explanation :

Suppose DATA is an array of N elements. Sorting these elements in ascending order means arranging the elements such that :

DATA [1] < = DATA [2] < = ... < = DATA [N]

In Bubble sort, compare DATA[1] with DATA[2] and exchange them if DATA[1] > DATA[2].

Next DATA[2] is compare with DATA[3]. They are exchanged if necessary. This process is repeated till DATA[N – 1] is compared with DATA[N].

One makes N – 1 comparisons, this is called a pass.

After the first pass the largest element is sink to the last position.

During the next pass, compare elements upto the last but one and second largest element moves to the $(N-1)^{st}$ position.

After N – 1 passes, all elements are sorted.

Consider a linear array consisting of 5 elements, given below :

Data[1]	55
Data[2]	43
Data[3]	05
Data[4]	06
Data[5]	09

Pass 1 :

(a) Compare DATA[1] with DATA[2] since 55 > 43 ∴ exchanged

(55) (43) 5 6 9

∴ New list is 43 55 5 6 9

(b) Next compare DATA[2] with DATA[3] since 55 > 5 ∴ exchanged

43 (55) (5) 6 9

∴ New list is 43 5 55 6 9

(c) Now, compare DATA[3] with DATA[4]since 55 > 6 ∴ exchanged

43 5 (55) (6) 9

∴ New list is 43 5 6 55 9

(d) Compare DATA[4] with DATA[5] since 55 > 9 ∴ exchanged

43 5 6 (55) (9)

∴ New list is 43 5 6 9 55

At the end of first pass, the largest element 55, has moved to the last position.

Pass 2 : In this pass, only three comparisons since K = 2.

(a) (43) (5) 6 9 55 Since 43 > 5 ∴ exchanged

∴ New list is 5 43 6 9 55

(b) 5 (43) (6) 9 55 Since 43 > 6 ∴ exchanged

∴ New list is 5 6 43 9 55

(c) 5 6 (43) (9) 55 Since 43 > 9 ∴ exchanged

∴ New list is 5 6 9 43 55

At the end of second pass, the second largest element 43 has maked to its proper position.

Pass 3 :

(5) (6) 9 43 55 Since 5 < 6 ∴ No exchange

∴ 5 (6) (9) 43 55 Since 6 < 9 ∴ No exchange

∴ New list is 5 6 9 43 55

Pass 4 :

In this way after complete execution of this algorithm, the array gets sorted in ascending order as follows :

DATA[1]	05
DATA[2]	06
DATA[3]	09
DATA[4]	43
DATA[5]	55

Q. 20 What do you understand by the term searching ? Which are the different types of searching algorithms ? Explain the linear searching algorithm. **(March 2004, Oct. 2004, 2010, 2021**

Ans. : Searching : Searching means to find out particular element from a given list of elements or check whether required element is present or not in an array. There are two types of searching algorithms as follows :

(1) Linear search (2) Binary search

Linear searching algorithm :

In linear search the given element is compared with each element of list one by one. For algorithm, refer to Q. No. 21.

Q. 21 Write an algorithm for linear search technique with suitable example. **(March 2003, 07, 09; Oct. 2007, 10, 21, July 2016, Dec. 2020**

Ans. :

Algorithm : Linear Search

LINEAR(DATA, N, ITEM, LOC)

Here DATA is a linear array with N elements and ITEM is given element. This algorithm finds the location LOC of ITEM in DATA or sets LOC = 0, if search is unsuccessful.

Step 1 : [Insert ITEM at the end of DATA]
Set DATA [N + 1] := ITEM

Step 2 : [Initialize counter]
Set LOC := 1

Step 3 : [Search for item]
Repeat While DATA [LOC] ≠ ITEM :
Set LOC := LOC + 1
[End of loop]

Step 4 : If LOC = N + 1, then :
Set LOC := 0

Step 5 : Exit

For example : Given DATA array with following 5 elements

11 22 33 44 55

Suppose ITEM = 33

Step 1 : Set DATA [6] = 33, List becomes

11 22 33 44 55 33

Step 2 : LOC = 1

Step 3 : Since DATA [1] = 11 ≠ 33 ∴ LOC = 2

Since DATA [2] = 22 ≠ 33 ∴ LOC = 3

Here DATA [3] = 33 = 33 = ITEM

Step 4 : Hence ITEM = 33 found at position, LOC = 3.

Q. 22 Write an algorithm for binary search technique with example (Oct. 2002,06,11,12,13 (Mar.2013, 14, 15,19, 22

Ans. : Binary search is used to search an element from sorted array.

Algorithm : Binary search

Binary (DATA, LB, UB, ITEM, LOC)

Here DATA is a sorted array with lower bound LB and upper bound UB. ITEM is given element. BEG denotes beginning, MID denoted middle and END denotes end location of DATA. This algorithm finds the location LOC of ITEM in DATA or sets LOC = NULL, if search is unsuccessful.

Step 1 : [Initialize Variables]

Set BEG : = LB, END := UB and MID := INT ((BEG + END)/2)

Step 2 : Repeat steps 3 and 4

while BEG = END AND DATA[MID] ITEM

Step 3 : If ITEM < DATA[MID], then :

set END := MID - 1

Else :

Set BEG := MID + 1

[End of If structure]

Step 4 : Set MID := INT ((BEG + END)/2)

[End of step 2 loop]

Step 5 : If DATA[MID] = ITEM, then :

set LOC := MID

Else :

LOC := NULL

[End of If structure]

Step 6 : Exit

e.g. Given DATA be the following sorted 13 element array :

11 22 30 33 40 44 55

60 66 77 80 88 99

Suppose ITEM = 40

Step 1 : Initially BEG = 1 and END = 13

Hence MID = INT[(1 + 13)/2] = 7

and so DATA[MID] = DATA [7] = 55

Step 2 : Since 40 < 55, END has its value changed by
END = MID – 1 = 7 – 1 = 6
Hence MID = INT [(1 + 6)/2] = 3
and so DATA[MID] = DATA[3] = 30

Step 3 : Since 40 > 30, BEG has its value changed by
BEG = MID + 1 = 3 + 1 = 4
Hence MID = INT [(4 + 6)/2] = 5
and so DATA[MID] = DATA[5] = 40

∴ Found ITEM in location LOC = MID = 5

Q. 23 Explain the advantages of binary search algorithm with a suitable example. State any two disadvantages or limitations of binary search. **(March 07, 19; Oct. 03**

Ans. :

Advantages of binary search algorithm :

(1) Binary search algorithm is efficient as the search scope gets reduced to half the size of the array, with each iteration.

(2) The number of comparisons required are approximately equal to $\log_2 n$ which are less than linear search.

(3) For example :

Given array data with 7-sorted elements :

11 22 30 33 40 44 55

Suppose ITEM = 40

Step I : Initially BEG = 1 and END = 7

∴ MID = (BEG + END) / 2 = (1 + 7) / 2 = 4

∴ DATA [MID] = DATA [4] = 33

Step II : Since 33 < 40, BEG is changed as BEG = MID + 1 = 4 + 1 = 5

∴ MID = (5 + 7) / 2 = 6

∴ DATA [MID] = DATA [6] = 44

Step III : Since 44 > 40, END has its value changed by END = MID – 1 = 6 – 1 = 5

∴ MID = (5 + 5) / 2 = 5

∴ DATA [MID] = DATA [5] = 40

∴ ITEM found at location 5 in array.

In above example, only two comparisons are required because at each iteration MID is calculated only one half is checked.

In the same example, for linear search, 5 comparison are required.

Disadvantages :

1) The given list must be sorted.
2) The access of list must be random means the middle element can be accessed.
3) At each iteration, middle entry calculation is required.

Q. 24 Write difference between Linear search and Binary search.

(Mar.2014, 2017, July 2017

Ans. :

Linear Search Binary Search	Binary Search
1. Linear search performs on unsorted list of elements as well as sorted list.	1. For binary search, the elements in array are stored in alphabetically or numerically in sorted manner.
2. Compare the desired element with all elements in an array until the match is found	2. Compare the value of midpoint with desired value. If the value is greater than midpoint value, the first half is checked, otherwise second half is checked until search is successful or interval is empty.
3. Insertion of an element in an array can be performed very efficiently when array is not ordered.	3. An insertion of a new element requires that many elements be physically moved to preserved order.
4. For large size of array, time required for this search is very large.	4. For large size of array, comparatively time required is less.
5. Time complexity is as follows : worst case : N comparison Best case : 1 comparison	5. Time complexity as follows : worst case : $\log_2 N$ comparison Best case : 1 comparison

Q. 25 What are pointer arrays ? (Oct. 2003,06; Mar. 2012,15,19, July 2016

Ans. :

i) An array is called pointer array, if each element of that array is a pointer.

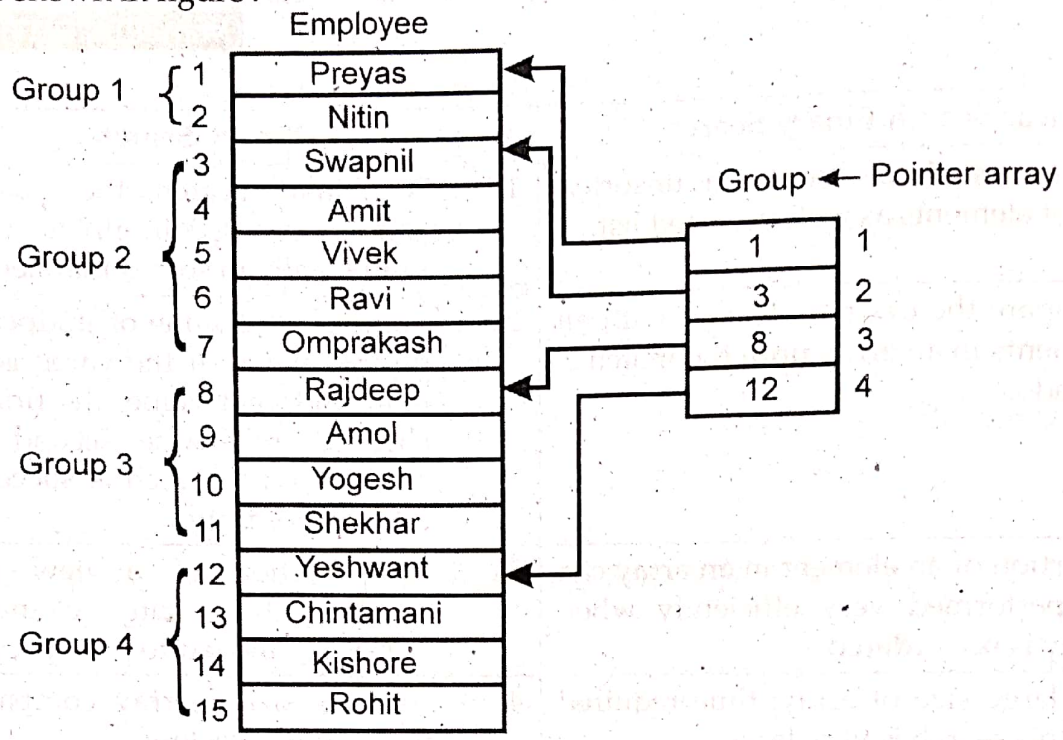
ii) The variable is called as pointer variable, if it points to another variable i.e. it contains the memory address of other variable.

iii) Consider an organization, which divides its employee list into four groups, depending on certain conditions. Following figure shows the list of 4 groups. There are 15 employes and groups contain 2, 5, 4 and 4 employes respectively as

Group 1	Group 2	Group 3	Group 4
Deepak	Swapnil	Rajdeep	Yashwant
Nitin	Amit	Amol	Chintamani
–	Vivek	Yogesh	Kishore
–	Ravi	Shekhar	Rohit
–	Omprakash	–	–

iv) If these groups are to be represented in memory, the most efficient way is to use 2 arrays. The first is Employee array, which contains list of employees in all four groups sequentially, while the second array is Group array, which is a pointer array, which contains the starting address of each group in the Employee array, respectively.

v) It is shown in figure :



vi) Each element of Group array is a pointer, which holds the starting addresses of different groups. Hence, it is called as pointer array.

Q. 26 What is a record ? (March 2011, 18,19

Ans. :

i) A record is a collection of relative data items, each of which is called as field or attribute.

ii) Collection of records is known as files. Collection of data is frequently organized into a hierarchy of fields, records and files.

iii) A record may contain non-homogeneous data i.e. data items of record need not to be of same data type. In a record, natural ordering of elements is not possible. The elements in record can be described by level number.

iv) e.g. An organization keeps records of its Employees. It contains following data items- Name, Sex, Salary, Birthday, Address.

Name is group item consisting of First name, Middle name and Last name. Also, Birth date and Address are group items.

The structure of this record is shown in figure below.

1. Employee
   2. Name
      3. First name
      3. Middle name
      3. Last name
   2. Sex
   2. Salary
   2. Birth date
      3. Date

3. Month

3. Year

2. Address

3. City

3. Pincode

v) The number to the left of each variable indicates level number.

vi) Employee (30)

This indicates a file of 30 records.

vii) To access first name of 3rd employee, we should write Employee (3).Name.First name.

In this way, we can access variables in records.

Q. 27 What is a record ? How it differs from a linear array ?

(March 2002, 05, 07, 08,14,16,22;Oct. 10,11; July 18, Dec. 2020

Ans. : A record is a collection of fields or attributes i.e. relative data items. Collection of data is frequently organized into hierarchy of fields i.e. records. A file is nothing but collection of records.

Difference between records and linear arrays :

(i) A record is a collection of fields, while an array is list of homogeneous data elements.

(ii) A record may contain non-homogeneous data i.e. data elements may be of different data types. An array always contains homogeneous data.

(iii) In a record, natural ordering of elements is not possible. Array elements can be naturally ordered.

(iv) Elements of record are referenced by level number, while those of array can be referenced by an index set consisting of n consecutive numbers.

Q. 28 How records are represented in memory using array ?

(Oct. 2002, March 2004, 18, Dec. 2020

Ans. :

i) Consider a record, whose structure is given below.

1. Employee

2. Name

3. First name

3. Last name

2. Sex

2. Address

3. City

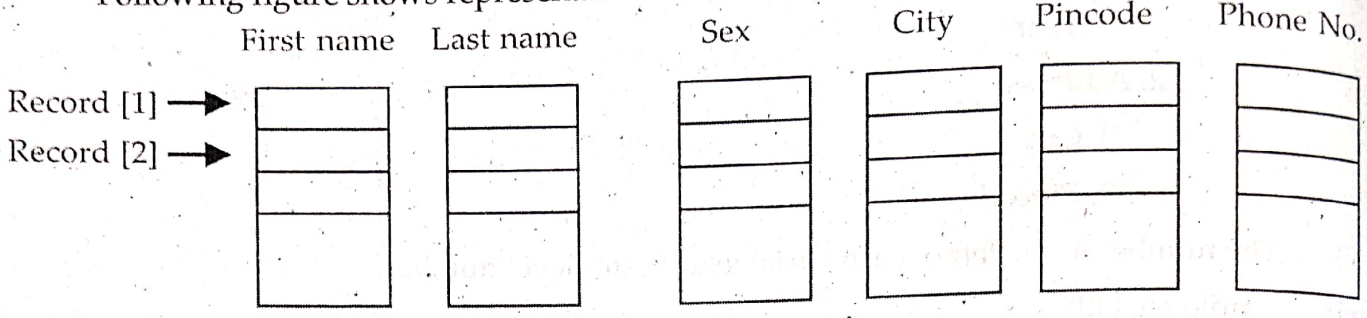
3. Pincode

2. Phone no.

ii) To represent this record in memory, linear arrays are used.

iii) One separate linear array is used for each elementary item of record such as First name, Last name, Sex, City, Pincode, Phone no.

Following figure shows representation of above record using parallel linear arrays.



iv) The records are stored in memory using parallel linear arrays, such that for an index K of all records, First name [K], Last name [K], Sex [K], ... belong to the same record in a file. (i.e. K^{th} record in the file)

Q. 29 Show representation of records in memory considering suitable example of three records and three fields. (Mar. 2003,11,13; Oct.2011,13

Ans. :

1) Records contain non-homogeneous data, so it cannot be stored in array.
2) But in entire file of records, all data elements belonging to the same identifier will be of same type. So a file may be stored in memory as collection of arrays.
3) One array for each of data item. All the arrays should be parallel.
4) For e.g.

A student file consisting three records and three fields.

Name	Address	Phone
Lokesh	11, J.M. Road	5662000
Jayesh	24, M.G. Road	4240020
Anushka	10, Sahkarnagar	4261900

Following figure shows representation of above file in three parallel arrays Name, Address and Phone –

Name
Lokesh
Jayesh
Anushka

Address
11, J.M. Road
24, M.G. Road
10, Sahkarnagar

Phone
5662000
4240020
4261900

All arrays should be parallel that is for subscript K the elements Name [K], Address [K], Phone [K] must belong to same record.

Linked List

Q. 30 What are linked lists ? Show a liked list with suitable example having six nodes with a properly labelled diagram.

(Mar. 2002,04,05,06,07,08,13,14,15; Oct. 2003,07,14, Dec. 202

OR

With suitable example, show labelled diagram for link between two nodes having the information part and next pointer field.

What are linked lists ? Show a liked list with suitable example having five nodes with a properly labelled diagram.

(Mar. 2013, March 2020

Ans. :

i) A linked list is a linear collection of data elements, called nodes, where the linear order is maintained with the help of pointers.

ii) Linked list is also called as one-way list.

iii) Each node in the linked list is divided into two parts. First part is called as INFO part, which contains the information about the element or actual element and second part is called as LINK part, which is next pointers field i.e. it contains the address of next node in the list.

iv) e.g.

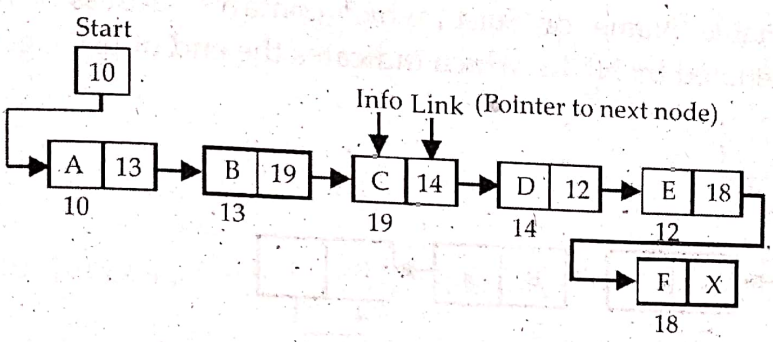


Fig. 1 : Linked list with 6 nodes

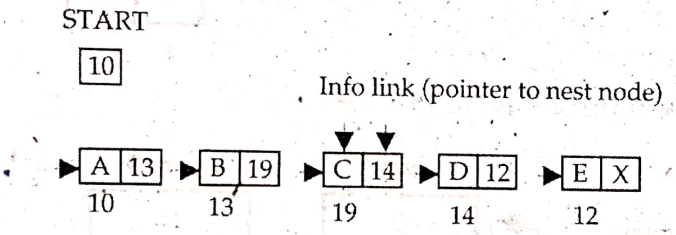


Fig. 2 : Linked list with 5 nodes

(a) The left part of the node is the Info part, which contains information of the element, while the right part is Link part, which is next pointers field i.e. it points to next node.

(b) An arrow is drawn from Link part of one node to the next node to indicate link.

(c) The address of the list is stored in Start or Name of the list.

(d) The Link part of last node is NULL pointer i.e. it contains nothing.

(e) To trace the linked list, we just require the address of Start or Name.

Q. 31 What are the advantages of linked lists over linear arrays ? **(March 2010)**

Ans. : Advantages of linked lists over arrays :

(i) To store arrays in memory, require consecutive memory locations, while to store linked lists, consecutive memory locations are not required.

(ii) Arrays can not be easily extended, while linked list can be easily extended.

(iii) There is very complicated procedure to insert an element in an array. One can easily insert an element in an linked list.

(iv) Similarly, deletion of an element from array is very complicated, while deletion from linked list is easy.

(v) Linked lists can be easily implemented and maintained in computer memory.

Q. 32 How linked lists are represented in memory ? OR

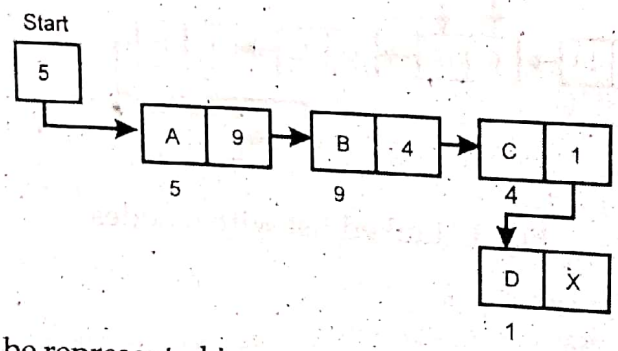
(March 2003,12,14,17,19,22; Oct. 2006, 07,13,21, July 16,18

With suitable example show the representation of linked list in memory.

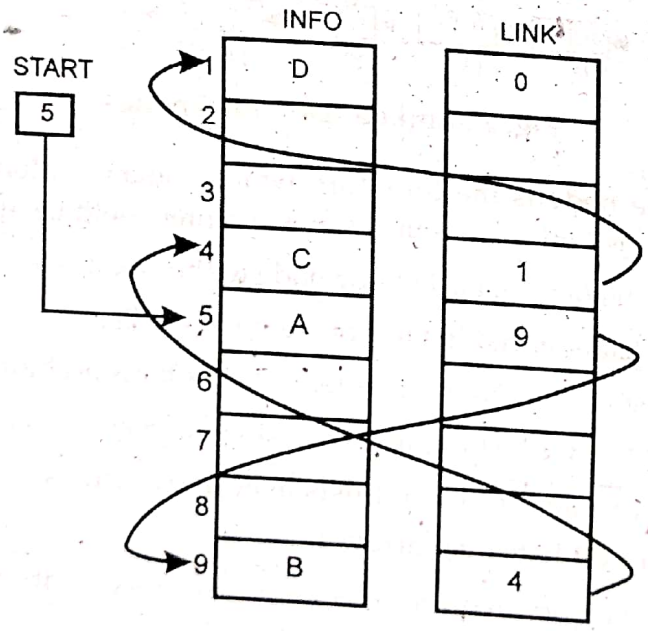
Ans. :

i) Linked lists can be represented in memory by using two arrays respectively known as INFO and LINK, such that INFO[K] and LINK[K] contains information of element and next node address respectively.

ii) The list also requires a variable 'Name' or 'Start', which contains address of first node. Pointer field of last node denoted by NULL which indicates the end of list. e.g. Consider a linked list given below :



iii) The linked list can be represented in memory as –



Above figure shows linked list. It indicates that the node of a list need not occupy adjacent elements in the array INFO and LINK.

Q. 33 Explain insertion and deletion from linked list with example.

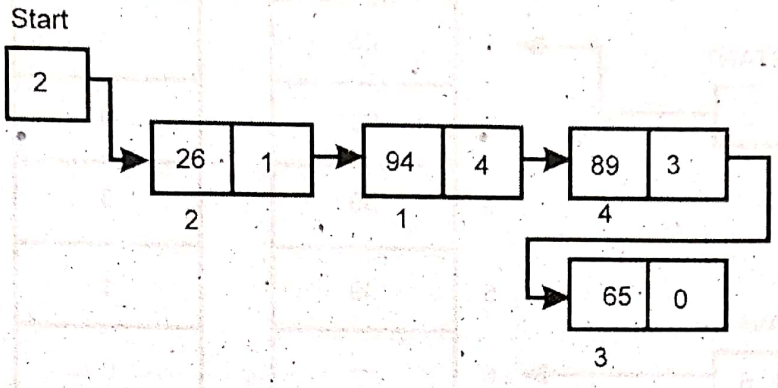
Ans. :

It is easier to insert an element into or delete an element from a linked list than arrays.

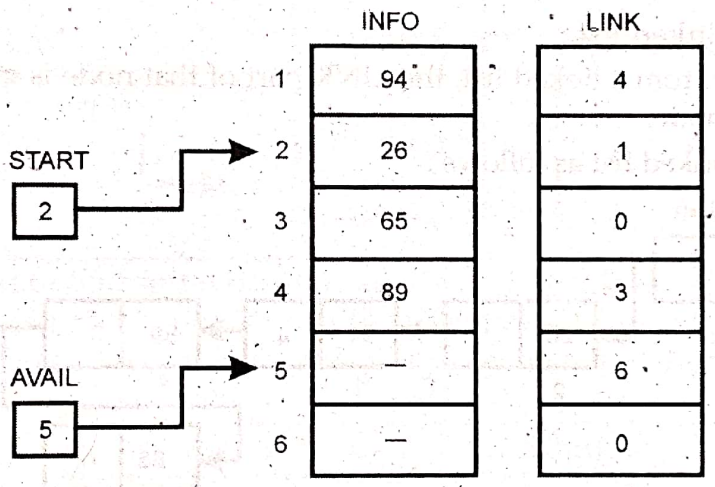
(i) Insertion into a linked list :

For insertion of an element into a linked list, the only requirement is that free memory space is available to store a node.

e.g. Consider a linked list having four nodes as follows.

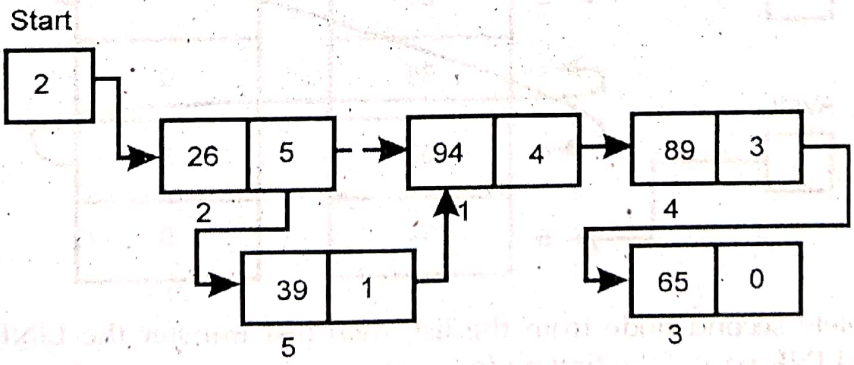


This list can be represented in memory as :

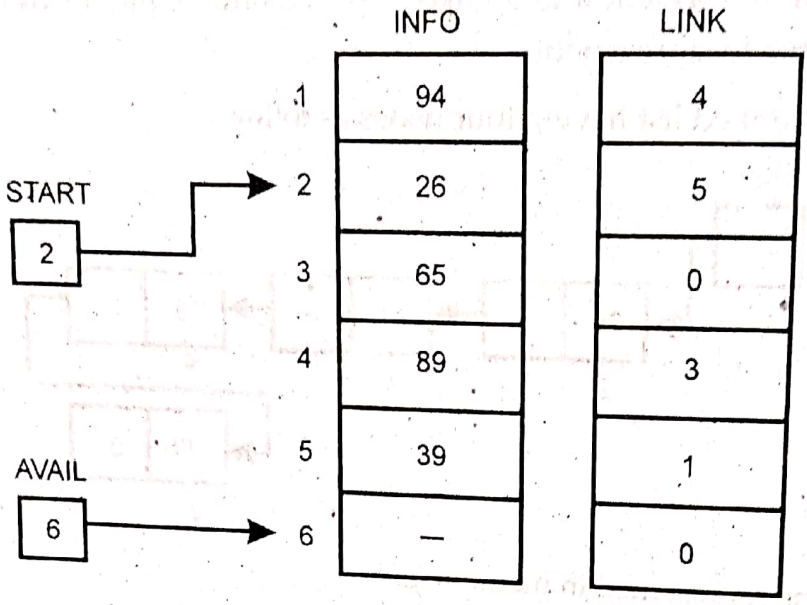


Now, to insert an element on second position of the list, the content of AVAIL are stored in LINK part of first node (since, AVAIL points to the memory location where new node can be inserted) and LINK part of the first node is transferred to LINK part of new node.

Then the list can be represented as follows.



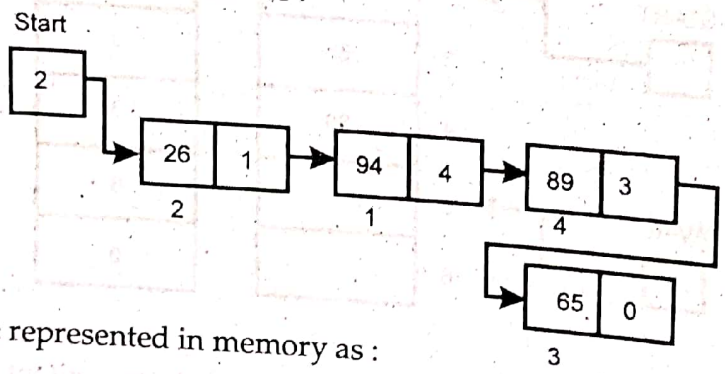
This list can be represented in memory as :



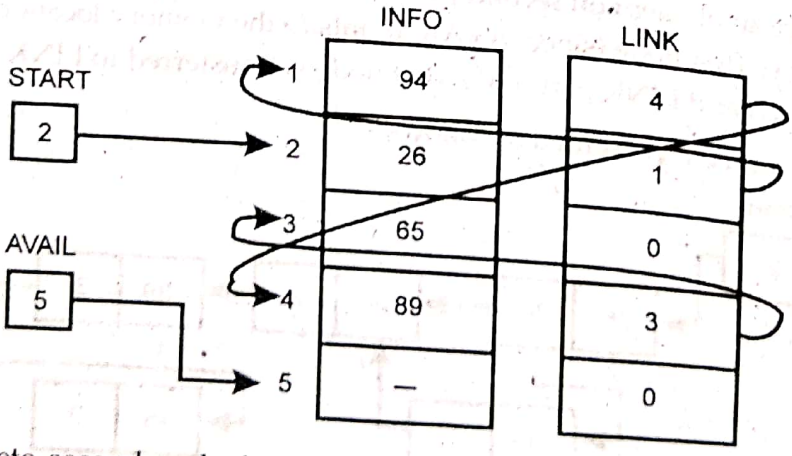
(ii) Deletion from linked list :

To delete a node from a linked list, the LINK part of that node is given to the LINK part of the previous node.

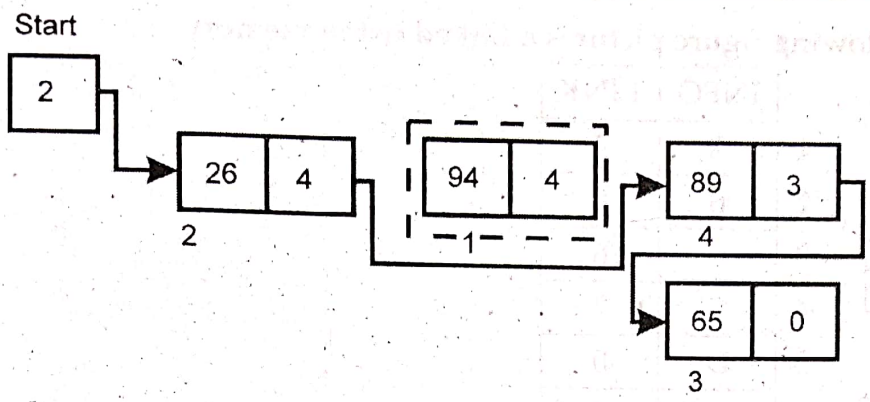
e.g. Consider a linked list as follows :



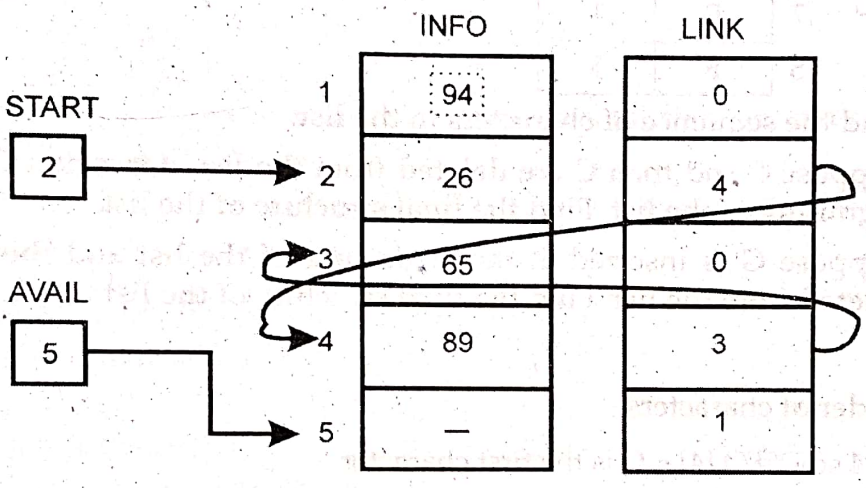
This list can be represented in memory as :



Now, to delete second node from the list, then just transfer the LINK part of second node to the LINK part of the first node.



This list can be represented in memory as



Q. 34 There is a list of 5 hospital patients and their corresponding room numbers. Fill the values of N start and N link so that they form an alphabetical link of patient names. Also fill the values of R start and R link so that they form an ordering of room numbers.

N start ☐

R start ☐

	Name	Room No.	N link	R link
1	Deepak	127		
2	Nitin	021		
3	Swapnil	420		
4	Amit	040		
5	Vivek	199		

Ans. :

N start 4

R start 2

	Name	Room No.	N link	R link
1	Deepak	127	3	5
2	Nitin	021	1	4
3	Swapnil	420	5	0
4	Amit	040	2	1
5	Vivek	199	0	3

Q. 35 The following figure pictures a linked list in memory.

Start: 4

Avail: 3

	INFO	LINK
1	A	2
2	B	8
3		6
4	C	7
5	D	0
6		0
7	E	1
8	F	5

(i) Find the sequence of characters in the list.

(ii) Suppose F and then C are deleted from the list. After that G is inserted at the beginning of the list. Find the final structure of the list.

(iii) Suppose G is inserted at the beginning of the list and then F after that C is deleted from the list. Find the final structure of the list.

Ans. :

i) Linear order of characters

START = 4 so INFO [4] = C is the first character

LINK [4] = 7 so INFO [7] = E is the second character

LINK [7] = 1 so INFO [1] = A is the third character

:

:

LINK [5] = 0, the NULL value so the list is ended.

∴ C E A B F D is the character string. Hence sequence is C, E, A, B, F, D.

(ii)

START: 3

AVAIL: 6

	INFO	LINK
1	A	2
2	B	5
3	G	7
4	(C)	0
5	D	0
6	–	8
7	E	1
8	(F)	4

Sequence

G, E, A, B, D

(iii)

START

	3

AVAIL

	6

	INFO	LINK
1	A	2
2	B	5
3	G	7
4	C	0
5	D	0
6		8
7	E	1
8	F	4

Sequence

G, E, A, B, D

Q. 36 Let LIST be a linked list in memory. Write an algorithm for traversing the linked list for following purposes :

(i) Find the number of times given ITEM occurs in the list.

(ii) Find number of non-zero elements in the list.

(iii) Add given value K to each element of the list.

Ans. : Algorithm : Traversing a linked list

1. Set Ptr := START
2. Repeat While Ptr ≠ NULL :
   Apply process to INFO[Ptr]
   Set ptr := LINK [ptr]
   [End of loop]
3. Exit

(i)
1. Set Ptr := START
2. Set N := 0
3. Repeat steps 4 and 5 While Ptr NULL :
4. If INFO [ptr] = ITEM, Then :
   set N := N + 1
   [End of If structure]
5. Set Ptr := LINK [ptr]
   [End of step 3 loop]
6. Write : N
7. Exit

(ii)
1. Set Ptr := START
2. Set N := 0
3. Repeat steps 4 and 5 While Ptr ≠ NULL :
4. If INFO [Ptr] ≠ 0, Then :
   Set N := N + 1
   [End of If structure]

5. Set Ptr := LINK [Ptr]
   [End of step 3 loop]
6. Write : N
7. Exit

(iii)
1. Set Ptr := START
2. Repeat While Ptr ≠ NULL :
   Set INFO [Ptr] := INFO [Ptr] + K
   Set Ptr := LINK [Ptr]
   [End of loop]
3. Exit

Stack and Queue

Q. 37 Explain Stack and Queue with suitable examples. OR Explain LIFO and FIFO Systems with suitable examples. (Mar.2013, July 2017) (Oct. 2005, 2010)

Ans. : LIFO System :

(i) LIFO system is last-in-first-out system. In this type of system, the element which is inserted at last, will be deleted first.

(ii) Stack is an example of LIFO system. It is a linear system in which insertion and deletion takes place only at one end i.e. top of the list.

(iii) The insertion operation is referred to as push and deletion operation as pop.
e.g. consider a stack of dishes. If we want to add a new dish to this stack then it is added at the top of stack also deletion takes place from the top.

FIFO System :

(i) A FIFO system is first-in-first-out system. In this type of system, the element which is inserted first in the list will also be deleted first.

(ii) Queue is an example of FIFO system. A queue is a linear list, in which insertion takes place only at one end of the list known as 'rear' of the list and deletion takes place at the other end, called as 'front' of the list.
e.g. A queue for tickets in a cinema hall.

Tree

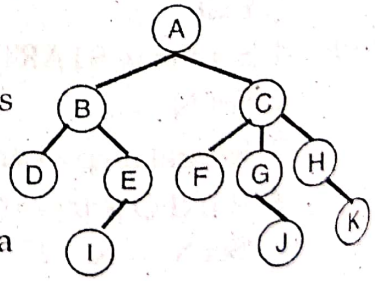
Q. 38 What is a tree ? What do you mean by root, leaf, siblings and child about tree. (Oct. 2006, 10, 21)

Ans. : Tree :

Tree is a non-linear hierarchical data structure which consists of finite set of one or more nodes (i.e. collected data items) such that :

a) There is specially designated node called the root.

b) The remaining nodes are partitioned into n≥ 0 finite disjoint sets T1, T2,...., Tn where each of these set is tree.
T1, T2, ..., Tn are called 'subtrees' of the root.
For e.g. figure shows tree which has 11 nodes, each item of data being a single letter.



Root :

A node which has no parent. Generally first node is called as 'root node'. In figure, a the node A is the root of the tree.

Leaf : (July 2018)

The node which has no child or children. Such nodes have degree zero. In figure a D, I, F, J, K are the leaf nodes. Also called as terminal node.

Child :

The nodes which are reachable from a node, say u, through a single edge are called the children of u. e.g In figure a, the children of node C are F, G, and H.

Sibling : (July 2018)

Children of the same parent are said to be siblings. e.g. The nodes D and E are both children of node B. So D and E are siblings.

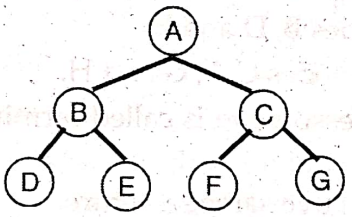
Q. 39 Explain the following terms : (March 2022)

Ans. : 1. Level of tree :

Each node in a tree is assigned a level number. Generally, the level number of root 'R' of the tree is zero and every other node is assigned to level number which is one more than the level number of its parent.

It is the distance from the root.

For e.g.



Level 0
Level 1
Level 2

2. Depth / Height : (March 2017)

Depth of a tree is defined as maximum level of any node in a tree. If root is level 0 then depth or height of tree is equal to 1 + largest level number.

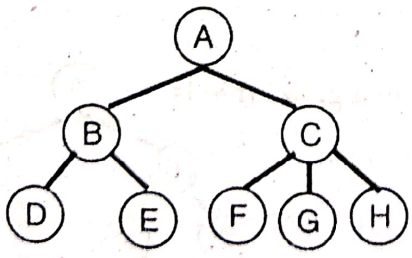
e.g. Depth of above tree is 3.

3. Degree :

The number of subtrees of a node is called degree of a node.

The degree of a tree is the maximum degree of the node in tree.

e.g. the degree of each node in figure is as



Node	Degree
A	2
B	2
C	3
D, E, F, G, H	0

The tree has degree 3.

Q. 40 What is a binary tree ?

(March 2002,04,05,14,15,19, 20; Oct. 2004,06,11,12,13; Dec. 202

Ans. : Binary tree is a finite set of elements called nodes such that is:

1. It may be empty or
2. It is partitioned into three disjoint subsets :

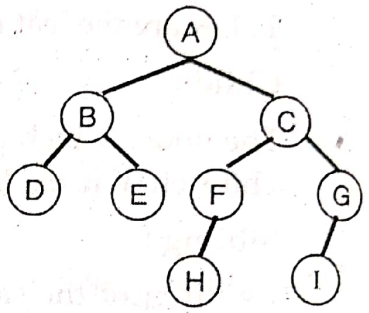
(a) there is a single distinguished element called the root of tree.

(b) other two subsets are themselves binary tree called left subtree and right subtree of the original tree.

A left and right subtree can be empty.

In binary tree, there is no node with degree greater than two.

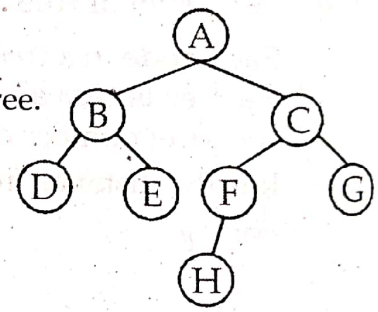
e.g.



Q. 41 What is a binary tree ? With a suitable example, explain the terminology describing family relationship between the elements of a binary tree. (March 2005, 11; July 18

Ans. : Binary tree :Please refer Q. No. 40.

Basic terminology :Consider the example :



The binary tree contains 8 nodes (A to H). Root A is at the top of tree.

(1) Left successor : B is left successor of node A.

(2) Right successor : C is right successor of node A.

(3) Left subtree : Left subtree consists of nodes B, D and E.

(4) Right subtree : Right subtree consists of nodes C, F, G and H.

(5) Terminal node : The node with no successors are is called terminal node D. E, H and G are terminal nodes.

(6) Binary tree T1 and T2 are similar if they have same structure.

Any node N in a binary tree T has either 0, 1 or 2 successors.

Q. 42 What is a binary tree ? With a suitable example show the relationship between total numbers of nodes and depth of a tree. (Oct. 2003,15, March 2006, 18; Dec. 2020

Ans. : Binary tree :Please refer Q. No. 40.

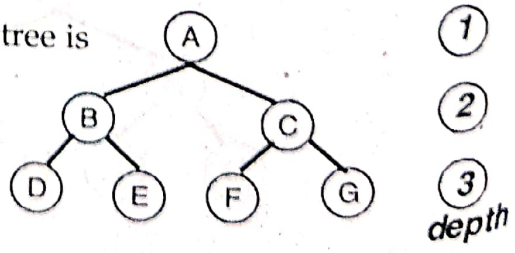
Relationship between total number of nodes and depth of a tree :

Depth of a tree means maximum level of any node in a tree. Maximum number of nodes of binary tree with depth n are $2^n - 1$.

For example :

Consider the following tree with depth 3.

So with depth 3, the total number of nodes in a given tree is



$2^n - 1 = 2^3 - 1$

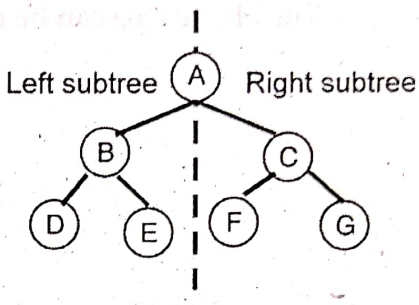
$= 8 - 1$

$= 7$

4. The tree with depth n having $2^n - 1$ number of total nodes.

Q. 43 Define the following :

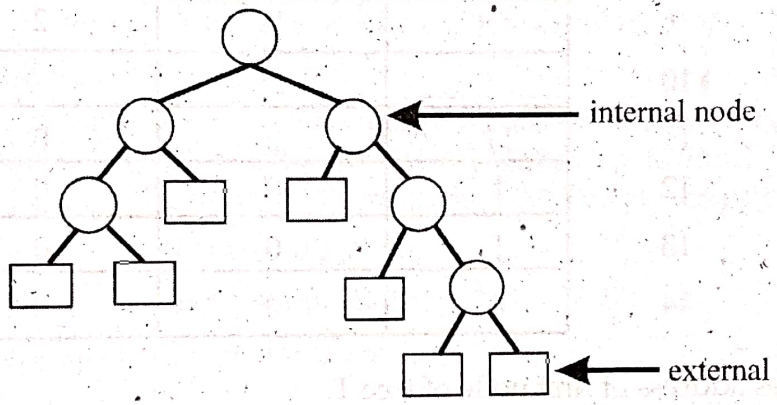
Ans. :



1. Complete Binary Tree :If all leaf nodes of a binary tree have same level number and every non-leaf node has non-empty left and right subtrees then the tree is called as complete binary tree. All nodes at the last level appears as far left as possible.

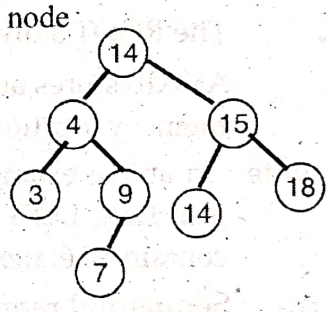
2. Extended binary tree or 2-tree : (Mar.2015)

A binary tree T is said to be a 2-tree or an extended binary tree if each node N has either 0 or 2 children. The nodes with 2 children are called internal nodes and the nodes with 0 children are called external nodes.



3. Binary Search Tree :

It is a binary tree in which each node N of tree has the property that the value at N is greater than every node value in the left subtree of N and is less than or equal to every node value in the right subtree of N.



Q. 44 How binary trees are represented in memory ? OR (Mar.2015, 20, July 2016, 19

With suitable example and labelled diagram, show the representation of binary tree in memory. (March 2003, 2009)

Ans. :

A binary tree T can be represented in memory by two types of representation :

(i) Linked representation

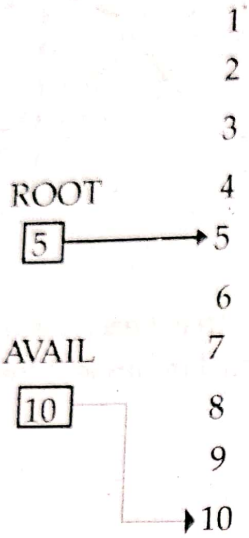
(ii) Sequential representation.

(i) Linked representation : (Oct. 2008,15)

Linked representation uses three parallel arrays INFO and, LEFT and RIGHT and a pointer variable ROOT such that for an index K, INFO [K] contains actual element, LEFT [K] contains address of left child and RIGHT [K] contains address of right child.

e.g. Consider a binary tree as below :

The above tree can be represented in memory as,



INFO	LEFT	RIGHT
B	7	14
G	12	13
	0	
E	11	0
A	1	9
	3	
D	0	0
	6	
C	14	2
H	0	0
I	0	0
J	0	0
F	0	0

The ROOT stores address of first node of tree T.

AVAIL stores address of first null node. To insert another node to tree T, it is inserted at memory location pointed by AVAIL.

Note : In above example, to insert an element K, then it will be inserted at INFO [10]. After insertion, LEFT [10] and RIGHT [10] will contain zero (null pointer) and AVAIL will contain 8 i.e. next element is to be inserted at 8.

(ii) Sequential representation :

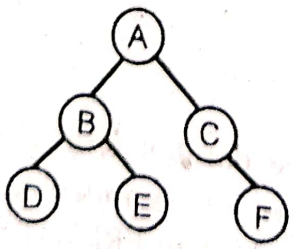
For sequential representation, only one linear array is used. This array is generally known as TREE such that :

(a) The root R of the tree is stored in TREE [1].

(b) If a node N of tree stored in TREE [K], then,

its left successor is stored in TREE [2*K] and right successor is stored in TREE [(2*K) + 1]

e.g. Consider a binary tree as follows :



This tree can be represented in memory as,

	TREE
1	A
2	B
3	C
4	D
5	E
6	–
7	F
8	–
9	–
10	–

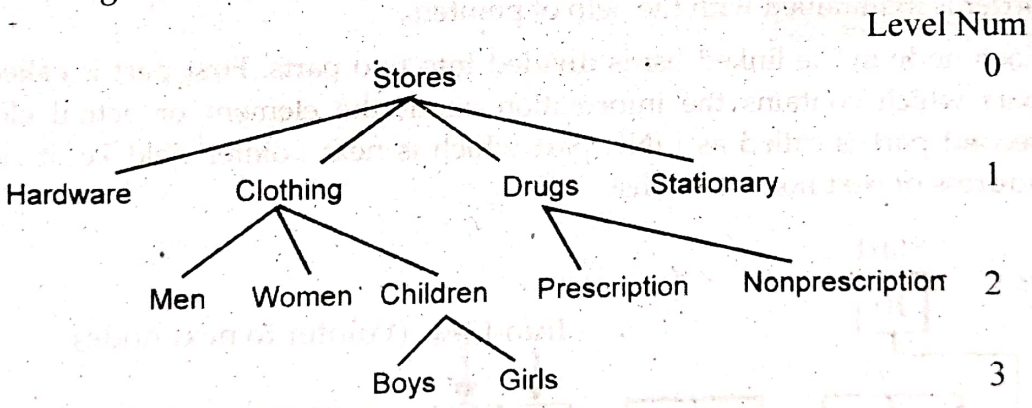
In general, the sequential representation of a tree with depth d will require an array with approximately 2^{d+1} elements.

Q. 45 Each store in a chain sends in a weekly record of its sales according to the following structure 01 Store, 02 Hardware, 02 Clothing, 03 Men, 03 Women, 03 Children 04 Boys, 04 Girls, 02 Drugs, 03 Prescription, 03 Nonprescription, 02 Stationary.

1. Draw the appropriate tree diagram.
2. How many elementary items are there ?
3. How many group items are there ?

Ans. :

1. The tree diagram is as follows :



2. Elementary items are those all nodes which have no children under given group. The above tree has 8 elementary items as :

   Hardware, Men, Women, Boys, Girls, Prescription, Nonprescription, Stationary.

3. Group items are those all nodes having children excluding root.

   The above tree has 3 group items as -

   Clothing

   Children

   Drugs.

Q. 46 Explain the following data structures with suitable diagram.

(a) Linear array (b) Linked list

(c) Tree

(March 2003,08,11,12,15,19; Oct. 2006)

Ans. :

(a) Linear array :

A linear array is the data structure which consists of finite ordered set of homogeneous data elements such that :

(i) The elements of the array are referenced respectively by an index set consisting on consecutive numbers.

(ii) The elements of the array are stored respectively in successive memory locations.

(iii) The number n of the elements is called length of size or array.

For e.g. let DATA be 5 elements linear array as follows :

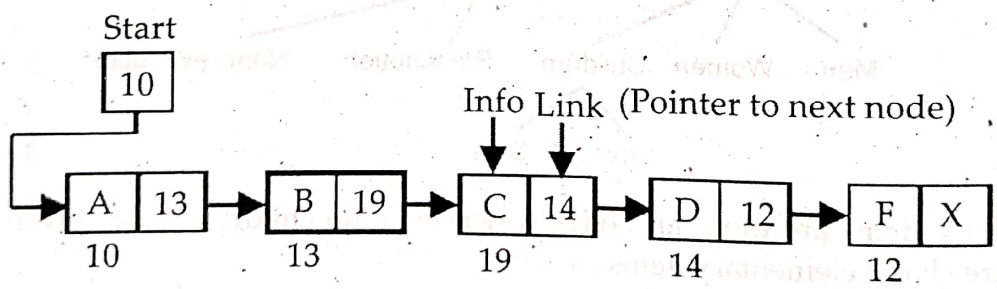
	DATA
1	300
2	400
3	100
4	50
5	09

(b) Linked list : (March 2019)

(i) A linked list is a linear collection of data elements called nodes where the linear order is maintained with the help of pointers.

(ii) Each node in the linked list is divided into two parts. First part is called as INFO part which contains the information about the element or actual element and second part is called as LINK part which is next pointer field i.e. it contains the address of next node in the list.

(iii)



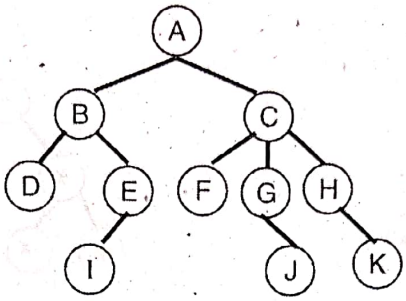
(c) Tree : (Oct. 2006)

Tree is a non-linear hierarchical data structure which consists of finite set of one or more nodes (i.e. collected data items such that :

(i) There is specially designated node called the root.

(ii) The remaining nodes are partitioned into n Z 0 finite disjoint sets T1, T2, ..., Tn where each of these set is tree.

T1, T2,....., Tn are called 'subtree' of the root.

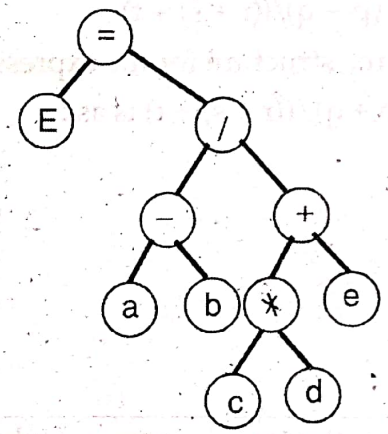


Q. 47 What are binary trees ? Draw the binary tree structure for the following expression : E = (a + b) / [(c * d) – e] (March 2004)

Ans. :

Binary Tree : Refer to Q. No. 40.

The binary tree structure for the expression E = (a + b) / [(c * d) – e]



Q. 48 Draw the tree structure for the following expressions. (March 2002,07,08; Oct. 2004,11

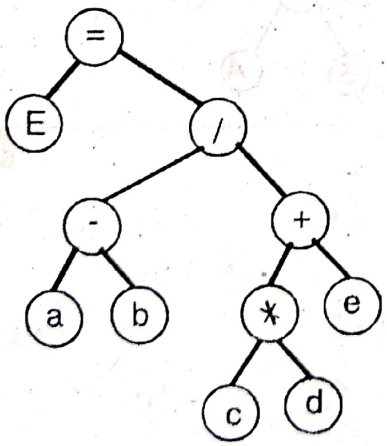
1) (i) E = (a – b) / (c * d) + e) (ii) E = (p – q) / (r * s) + t

2) [(a + b) * c]/[a * ((b – c) + a)]

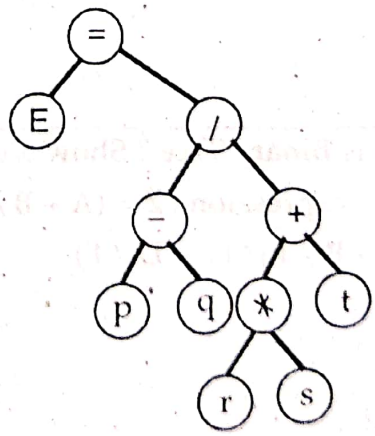
3) $(2x + y)(a - 7b)^3$

Ans. : 1. E = (a - b)/(c* d) + e)

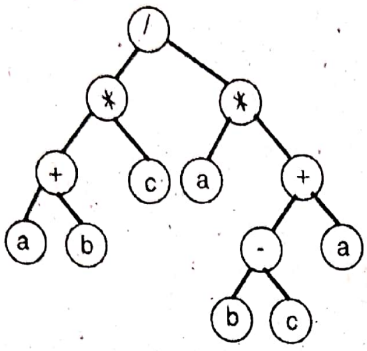
(i)



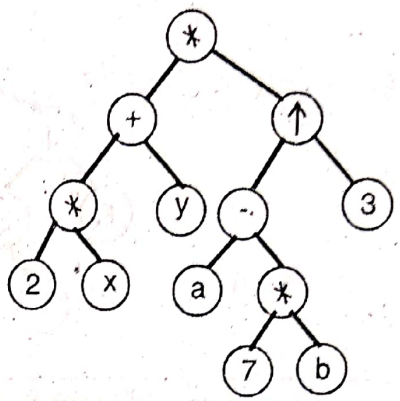
(ii) E = (p – q) / ((r * s) + t) (March 2022)



2. $[(a + b) * c]/[a * ((b - c) + a)]$



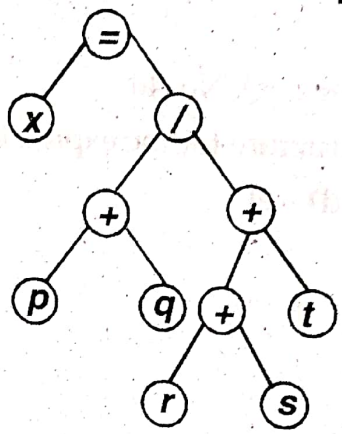
3. $(2x + y)(a - 7b)^3$ (March 2008)



Q. 49 Draw a tree structure for the following expression : (March 2008, Oct. 2005)

$X = (p + q)/((r + s) + t)$

Ans. : The tree structure for the expression $X = (p + q)/((r + s) + t)$ is as :

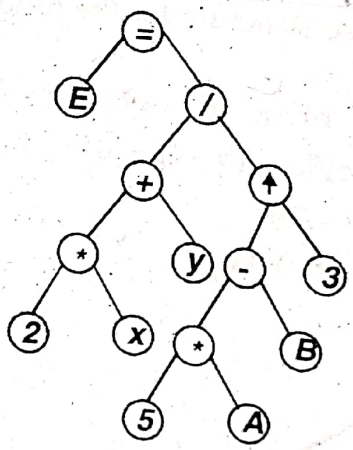


Q. 50 Draw the tree diagram which corresponds to the following algebraic expression :

$E = (2X + Y)/(5A - B)^3$

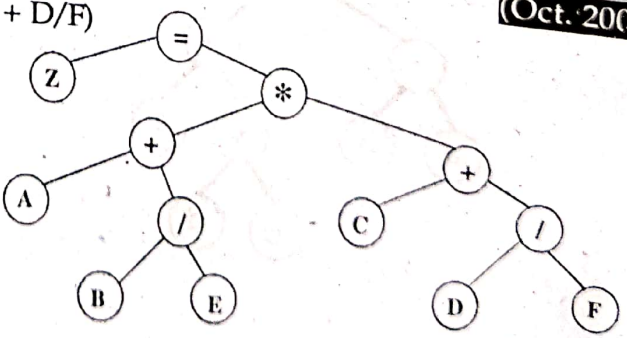
Ans. : Tree diagram which corresponds to the following algebraic expression. (March 2006)

$E = (2X + Y)/(5A - B)^3$ is as :



Q. 51 What is Binary Tree ? Show a tree structure for the expression : $Z = (A + B / E) * (C + D/F)$ (Oct. 2007)

Ans. : $Z = (A + B / E) * (C + D / F)$



Q. 52 Explain the terms Siblings and Leaf in case of a tree structure
Draw tree diagram for the expression
$C = (X/Y)^3 - (5A + B^2)$

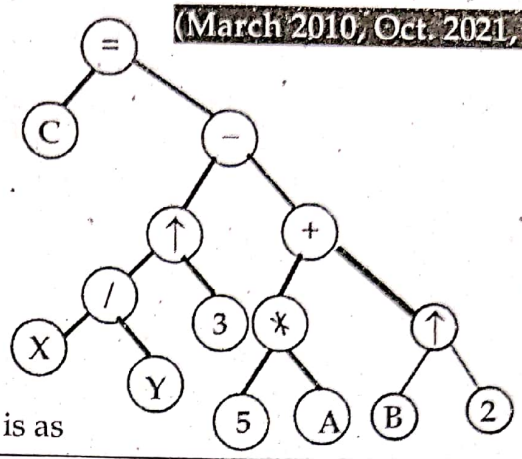
(March 2010, Oct. 2021, 3)

Ans. :

sibling & Leaf :

(Please Refer Ch2/Q-37/p-2-29)

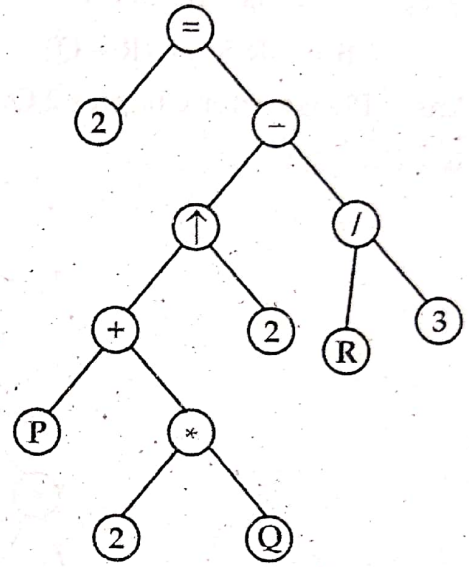
Tree diagram for the expression is as



Q. 53 What is a Tree Structure ? Draw the diagram for the given expression :
$A = (P + 2\ Q)^2 - (R/3)$

(Please refer Ch.2, Q.37 (Page No. 2 -29)

(Oct. 2010, 3)



Q. 54 Draw Binary Tree structure for the following expression :
$(2A + B)\ (5F - D)^3$

(March 2013, 4)

Ans. :

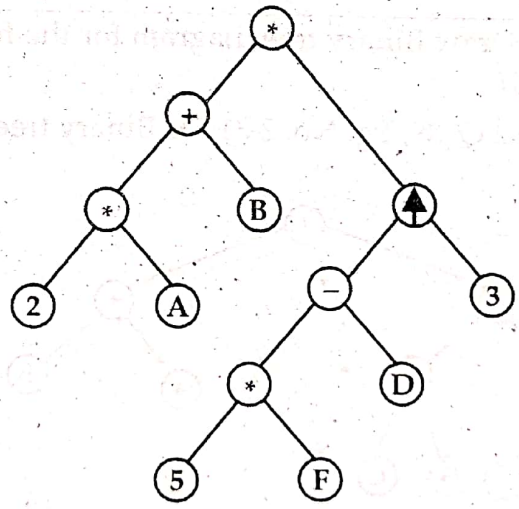


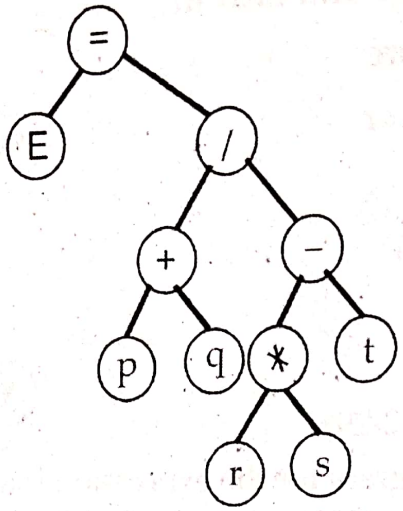
Fig. Q. 54

Q. 55 What is Binary Tree ? Draw the tree structure for the following expression
$E = (p + q) / [(r * s) - t]$

(Oct. 2012, 4)

Ans. :

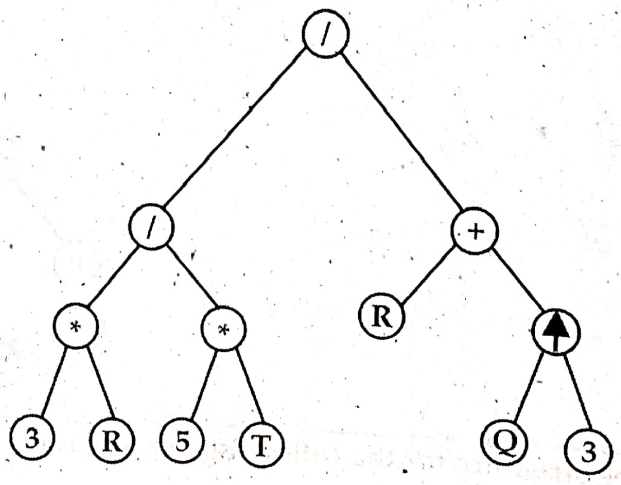
E = (p + q) / [(r * s) – t]



Q. 56 What is a Binary Tree? Draw tree diagram for the expression

$B = (3R/5T) - (R + Q^3)$ (Oct. 2013, 4

Ans. : Please refer Chapter 2 Q. 39, Pg. No. 2-30.

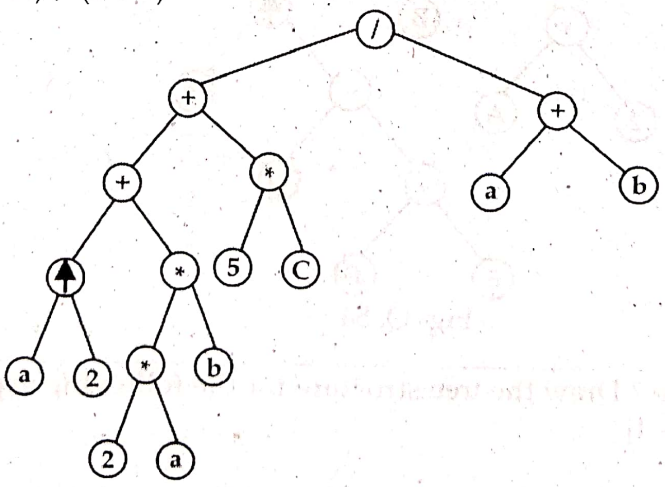


Q. 57 Define Binary Tree. Draw Binary tree diagram for the following expression :

$(a^2 + 2ab + 5c) / (a + b)$

Ans. : Please refer Chapter 2 Q. 39, Pg. No. 2-31 for Binary tree. (March 2014, 3

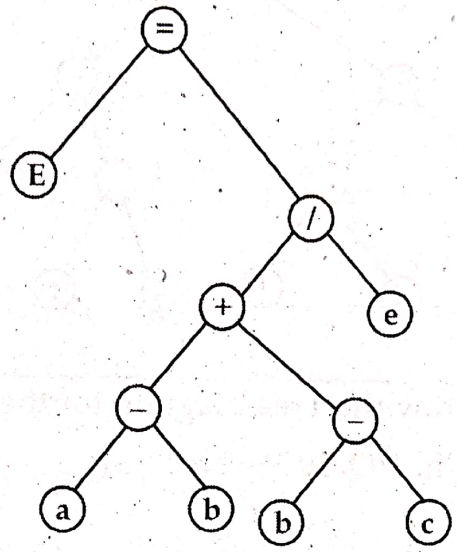
$(a^2 + 2ab + 5c) / (a + b)$



Q. 58 Draw the binary tree for following expression :
E = ((a – b) + (b – c)) / e

(Oct. 2014, 3)

Ans. :

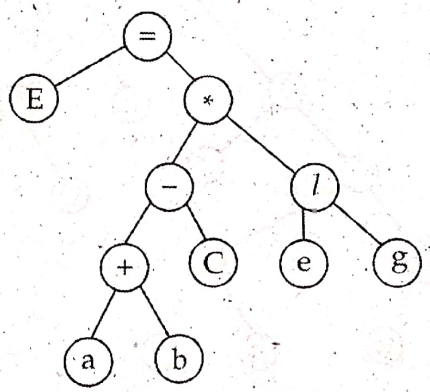


Q. 59 Draw the binary tree for following expression :
E = ((a + b) – c) * (e/g)

(Oct. 2015, 4)

Ans. :

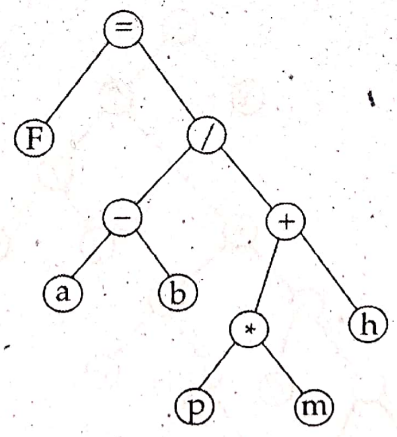
E = ((a + b) – c) * (e/g)



Q. 60 Draw a binary tree structure for the following expression :
F = (a – b) / ((P * m) + h)

(July 2017)

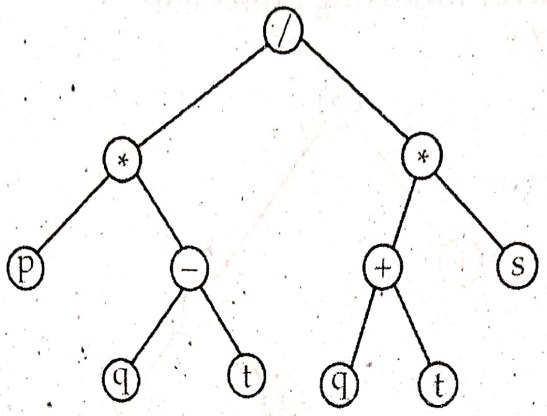
Ans. :



Q. 61 Draw a binary tree structure for the following expression :
(p * (q – t)) / ((q + r) * s)

(Oct. 2006)

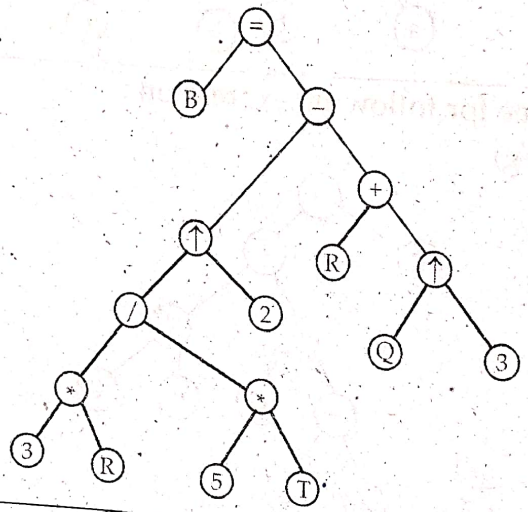
Ans. :



Q. 62 What is Binary Tree ? Draw the Tree diagram for the expression.

$B = (3R/5T)^2 - (R + Q^3)$ (Ch. 2/Q. 40/Pg. No. 2-31) (March 2019)

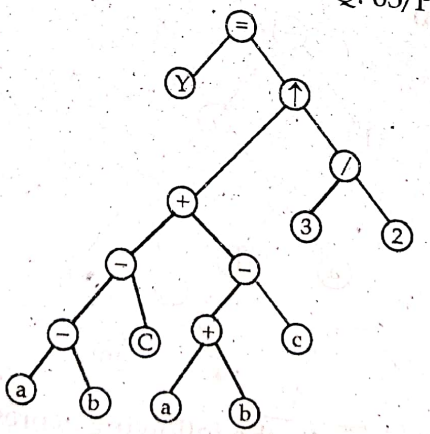
Ans. :



Q. 63 Define Binary tree. Draw a Tree diagram for following expression. (March 2020)

$Y = [(a - b - c) + (a + b - c)]^{3/2}$ (Ch. 2/Q. 40 and Q. 63/Pg. No. 2-31 and 2-41)

Ans. :



Q. 64 If symmetric binary tree contains 31 nodes then calculate in level of tree and its depth. (March 2022)

Ans. :

$$2^n - 1 = 31$$

$$2^n = 31 + 1$$

$$2^n = 32$$

$$n = 5$$

depth = 5 and Level = 4

(As depth = 1 + largest level number)

Q. 65 Select the correct alternative and rewrite the following statements

1. Finding location of given element is called as ——. (March 2005)

(i) Traversing (ii) Insertion

(iii) Searching (iv) None of the above

Ans. : (iii) Searching

2. Data items that are divided into subitems are called as — (March 2018)

(i) Group items (ii) Elementary items

(iii) Nodes (iv) Arrays

Ans. : (i) Group items

3. In LINKED LIST, Link field contains ——.

(i) Value of next node (ii) Address of next node

(iii) Value of previous node (iv) None of these

Ans. : (ii) Address of next node

4. A record is a collection of ——. (March 2017)

(i) Files (ii) Arrays (iii) Fields (iv) Maps

Ans. : (iii) Fields

5. The time required to execute bubble sort algorithm having 'n' input items is directly proportional to ——. (March 2013)

(i) n^2 (ii) n (iii) $Log_2 n$ (iv) $Log_e n^2$

Ans. : (i) n^2

6. Maximum number of nodes of symmetric binary tree with depth n are ——. (Oct. 2005, 09, March 2012)

(i) 2^n (ii) $Log_2 n$ (iii) n^2 (iv) $2^n - 1$

Ans. : (iv) $2^n - 1$

7. Maximum number of nodes of symmetric binary tree with depth 5 are ——. (March 2002)

(i) 5 (ii) 25 (iii) 31 (iv) 32

Ans. : (iii) 31

8. Accessing each element in an array only once is called ——. (Oct. 2002, March 03, 11)

(i) Searching (ii) Inserting

(iii) Deleting (iv) Traversing

Ans..: (iv) Traversing

9. The elements of record are ——. (Oct. 2003, March 2009)

(i) Homogeneous (ii) Similar
(iii) Non-homogeneous (iv) Identical

Ans. : (iii) Non-homogeneous

10. The most efficient search algorithm is ——. (March 2004)

(i) Binary search (ii) Reverse search
(iii) Linear search (iv) Pointer search

Ans. : (i) Binary search

11. The number of comparisons required for bubble sorting of an array of n elements is ————. (March 2004)

(i) $n(n-1)/2$ (ii) $n/2$
(iii) $\log_2 n$ (iv) $\log_{10} n$

Ans. : (i) $n(n-1)/2$

12. Finding the location of record with a given key value is known as ——. (March 05; Oct. 11)

(i) Traversing (ii) Searching
(iii) Sorting (iv) Inserting

Ans. : (ii) Searching

13. Maximum number of nodes in a symmetric binary tree with depth four are ——. (March 2006)

(i) 4 (ii) 15 (iii) 16 (iv) 5

Ans. : (ii) 15

14. In —— data structure, an element may be inserted or deleted only at one end called Top.

(i) Queue (ii) Array (iii) Stack (iv) Tree

Ans. : (iii) Stack

15. Maximum number of nodes of symmetric binary tree with depth of 6 is ——.

(i) 64 (ii) 6 (iii) 63 (iv) 36

Ans. : (iii) 63

16. _____ is the only non-linear data structure from the following list. (March 2022)

(i) Array (ii) Stack
(iii) Tree (iv) Linked List

Ans. : (iii) Tree

17. _____ is the operation of rearranging the elements of an array either in increasing and decreasing order. (March 2007)

(i) Sorting (ii) Searching (iii) DMS (iv) DBMS

Ans. : (i) Sorting

18. The complete binary tree (Tn) has n = 15 nodes then its depth (dn) is _____. (Oct. 2007)

i) 2 ii) 3 iii) 4 iv) 5

Ans. : (iii) 4

19. Maximum number of nodes of symmetric binary tree with depth of 7 is _____ (March 2008)

i) 125 ii) 127 iii) 128 iv) 124

Ans. : (ii) 127

20. Elements of Array are always _____. (Oct. 2008)

(i) Homogenous (ii) Hetrogenous

(iii) Non-homogenous (iv) None of these

Ans. : (i) Homogenous

21. Record contains _____ Data. (March 2009)

i) Homogeneous ii) Non-homogeneous iii) Same iv) None of these

Ans. : (ii) Non-homogeneous

22. Sorted List is essential requirement for _____ process of an array. (March 2010)

(i) Linear Search (ii) Binary Search (iii) Traversing (iv) Insertion

Ans. : (ii) Binary Search

23. Maximum number of nodes of symmetric binary tree with depth 6 are _____ (Oct. 2010)

i) 31 ii) 127 iii) 63 iv) 64

Ans. : (iii) 63

24. Tree is _____ Data Structure. (Oct. 2012)

(i) Linear (ii) Non-linear

(iii) Homogeneous (iv) Non-homogeneous

Ans. : (iv) Non-linear

25. The elements of the binary tree are _____. (Oct. 2013)

(i) Homogenous (ii) Non-homogeneous

(iii) Similar (iv) Identical

Ans. : (ii) Non-homogeneous

26. Complete Binary Tree (T_n) has n= 31 nodes, then its depth is _____. (March 2014)

(i) 2 (ii) 3 (iii) 4 (iv) 5

Ans. : (iv) 5

27. Most efficient search algorithm is_____. (March 2014)

(i) Binary (ii) Reverse (iii) Linear (iv) Pointer

Ans. : (i) Binary

28. Finding location of given element in array is called _____. (Oct. 2014)

(i) Sorting (ii) Searching

(iii) Traversing (iv) Merging

Ans. : (ii) Searching

29. _____data structure does not require contiguous memory allocation. (March 2015; Oct. 2021)

(i) Array (ii) String (iii) Pointer array (iv) Linked list

Ans. : (iv) Linked list

30. Tree is a _____ collection of Nodes. (Oct. 2015)

(i) Hierarchical (ii) Linear (iii) Relational (iv) Graphical

Ans. : (i) Hierarchical

31. If a complete binary tree (Tn) has n = 1000 nodes, then its depth (Dn) is _____. (July 2016)

(i) 10 (ii) 20 (iii) 50 (iv) 100

Ans. : (i) 10

32. _____ is the only non-linear data structure from the following list. (July 2017)

(i) Array (ii) Stack (iii) Tree (iv) Linked List

Ans. : (iii) Tree

33. If lower bound = 0 and upper bound = 15, then midterm is _____ in binary search method. (July 2018)

(i) 6 (ii) 7 (iii) 8 (iv) 9

Ans. : (iii) 8

34. _____ is very useful in situation when data is to be stored and retrieved in reverse order. (March 2019)

(i) Stack (ii) Queue (iii) Linked List (iv) Tree

Ans. : (i) Stack

35. Record contains _____ data. (July 2019)

(i) Homogenous (ii) Non-homogenous
(iii) Same (iv) None of these

Ans. : (ii) Non-homogeneous

36. _____ is collection of fields. (March 2020)

(i) File (ii) Record
(iii) Array (iv) Queue

Ans. : (ii) Record

37. In binary search method, the condition is data must be _____. (Dec. 2020)

(i) Sorted (ii) Unsorted
(iii) Random (iv) Discrete

Ans. : (i) Sorted

□□□