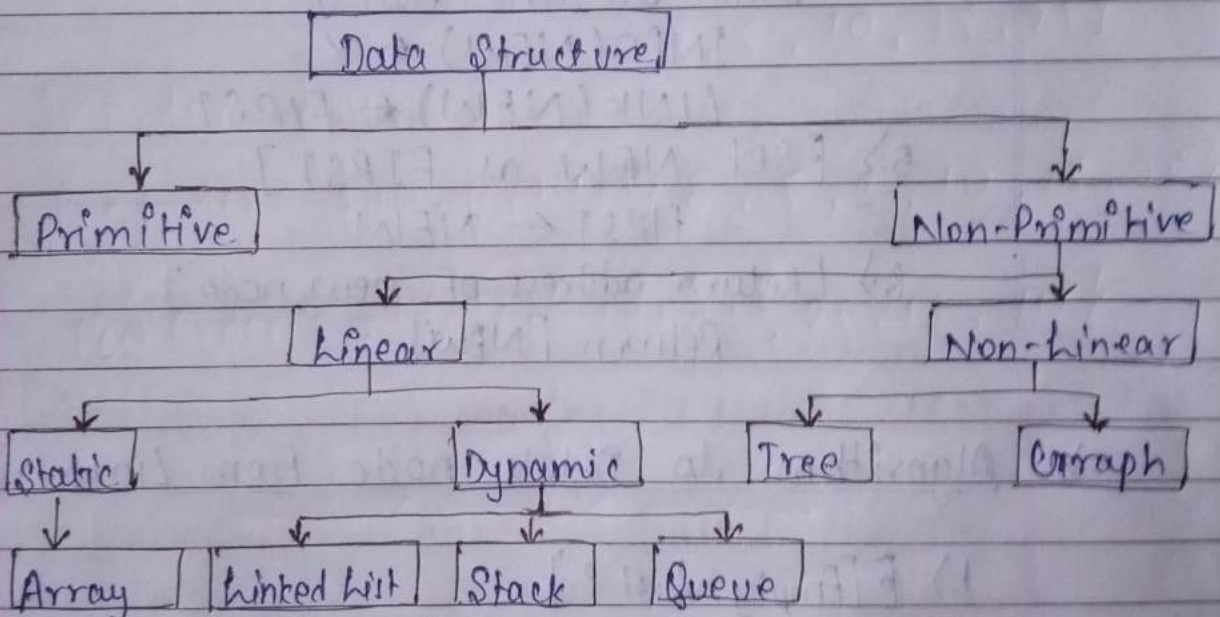
Name :- Aman kumar Singh
EN :- 20051010115 9
Batch :- C
BCA

1.) What is data Structure ? Explain its bifurcation ?

Ans Data Structure is a Programming way of Storing and Organizing data.
In simple words we can say "Way we organise our data", is called Data Structure.



2.) What in linked list ? Write algorithm to insert (Beginning) and delete element from linked list (singly)

Ans Linked List can be defined as collection of objects called Nodes that are randomly stored in memory.

=> Algorithm to insert node at beginning

1.) [Underflow ?]
IF AVAIL = NULL
then write ('Availability Stack Overflow')
Return (First).

2.) [Obtain address of next free node]
New ← Avail

3.) [Remove free Node from Availability Stack]
AVAIL ← LINK [AVAIL]

4.) [Intialise fields of new node and its link to the list]
INFO(NEW) ← X
LINK (NEW) ← FIRST

5.) [Set NEW as FIRST]
FIRST ← NEW

6.) [Return address of new node]
Return [NEW]

=> Algorithm to Delete node from Linked-List

1.) [Empty List ?]
IF FIRST = NULL
then Write ("Underflow")
Return.

2.) [Intialise Search for X]
TEMP ← FIRST

3.) [FIND X]
Repeat through step 5 while Temp ≠ X
and LINK (TEMP) ≠ NULL

4.) [Update predecessor / previous marker]
PRED ← TEMP

5.) [Move to next Node]
TEMP ← Link[Temp]

6.) [End of the List?]
if TEMP ≠ X
then Write ('Node not found')
Return

7.) [Delete X]
IF X = FIRST (Is X the first node?)
then FIRST ← LINK(FIRST)
else LINK(PRED) ← LINK(X)

8.) [Return node to availability area]
LINK(X) ← AVAIL
AVAIL ← X
Return.

3.) Write a note on stack with its operation (Algorithms)

Ans Stack are also an ordered collection of elements like array, but having a special feature that deletion and insertion of elements can be done only from one end called the Top.

=> Terminology in stack are :-
(i) Top
(ii) Bottom
(iii) Stack Overflow
(iv) Stack Underflow

=> Operation's in stack are :-
(i) PUSH
(ii) POP
(iii) PEEP
(iv) CHANGE

⇒ Algorithm for PUSH() operation

1.) [Check for Overflow]
if Top >= N then
write ("Stack Overflow")
Exit
2.) [Increment the Top pointer]
Top ← Top + 1
3.) [Insert an element into the stack]
stack [Top] ← item
4.) [Finished]
Return.

⇒ Algorithm for POP() operation

1.) [Check for stack underflow]
if Top == 0 then
write ("Stack Underflow")
Exit
2.) [Accessing the value to be deleted]
Item ← Stack [Top]
3.) [Decrement the Stack pointer]
Top ← Top - 1
4.) [Return deleted element Item]
Return Item.

⇒ Algorithm for PEEP() operation

1.) [Check for Underflow]
if (Top - i + 1) < 0 then
write ('stack Underflow')
Exit.
2.) [Storing the ith element in item]
item ← stack [Top - i + 1]

3.) [Finished]
Return item

4.) Write a note on Circular Queue with its operation (Algorithms)

Ans In Circular queue any elements in accessible from any position but only in a forward manner.

=> Algorithm to insert an element in ~~Double~~ Circular Queue.

1.) [Reset rear Pointer ?]
if R = N
then R <- 1
Else
R <- R+1

2.) [Check for Overflow ?]
if R = F
then write ('Overflow')
Return

3.) [Insert Element]
Q[R] <- y

4.) [Is front pointer properly set ?]
if F = 0
then F <- 1
Return

=> Algorithm to delete an element in Circular queue

1.) [Underflow ?]
    if F = 0
        then write ('Underflow')
            Return.

2.) [Delete an Element]
    Y ← Q[F]

3.) [Queue empty]
    if R = F
        then R ← F ← 0
    Return (Y)

4.) [Increment front pointer]
    if F = N
        then F ← 1
    else
        F ← F + 1
    Return (Y)

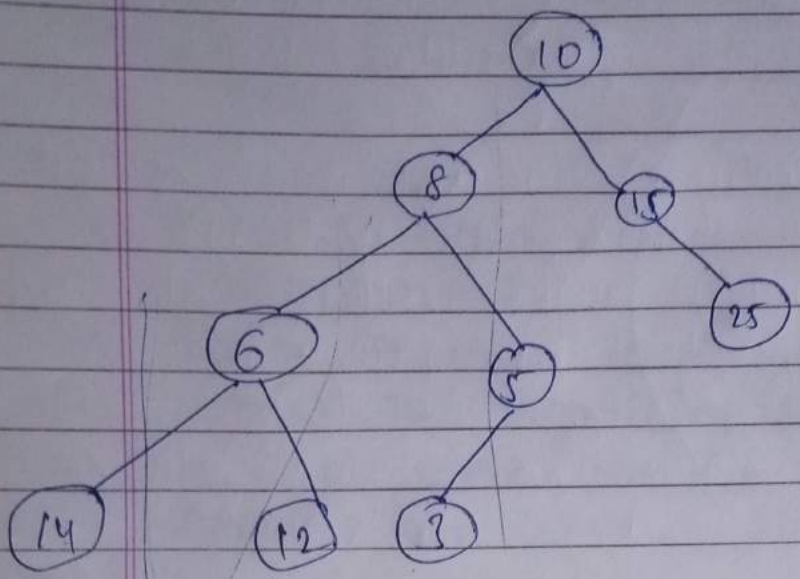
5.) What is Tree ? Explain Tree Traversal with example.

Ans A Tree is also one of the data structure that represent hierachical data.

Tree transversal means transversing or visiting each node of a Tree.
3 ways of Transversal are :-

(i) InOrder transversal (LRR)
(ii) PreOrder transversal (RLR)
(iii) PostOrder transversal (LRR)

Let's understand with an example



In Order = {14, 6, 12, 8, 3, 5, 10, 15, 25}

Pre Order = {10, 6, 14, 12, 8, 5, 3, 10, 15, 25}

Post Order = {14, 12, 6, 3, 5, 8, 25, 15, 10}