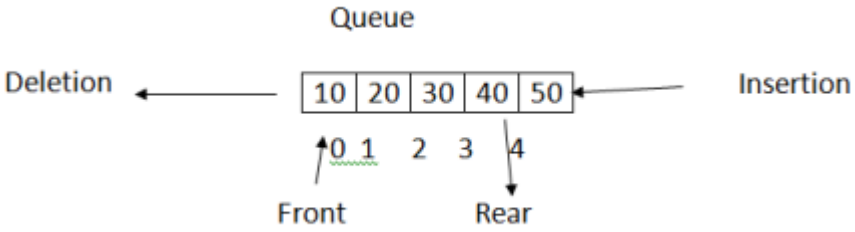
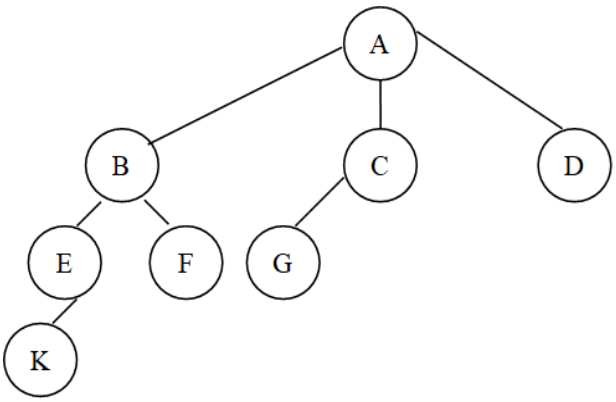
SUMMER – 2023 EXAMINATION
Model Answer – Only for the Use of RAC Assessors

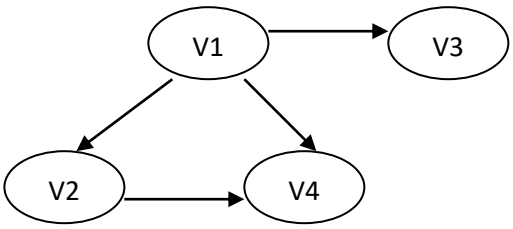
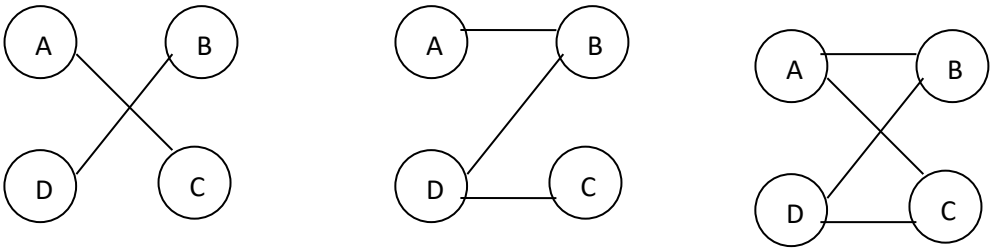
Subject Name: Data Structure Using C **Subject Code:** 22317

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.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**Write any four operations performed on data structure.**	**2 M**
	Ans	Insert Travese Create Destroy Select Update Copy Merge Search Sort	Each one carries ½ M

	b)	**Draw the diagram of Linear Queue to represent front and rear pointers.**	**2 M**
	Ans		Correct diagram carries - 2 M
	c)	**State the following terms:** **(i) Leaf node of a tree** **(ii) Degree of a tree**	**2 M**
	Ans	**Leaf Node:** Lead node is a terminal node of a tree. It does not have any nodes connected to it. K, F, G, and D are leaf nodes. All other nodes are called non-leaf nodes or internal nodes **Degree of the Tree:** The degree of a tree is the maximum degree of the nodes in the tree. The degree of the shown tree is 3. 	Each definition with example carries – 1 M
	d)	**Write any two operations performed on the stack.**	**2 M**
	Ans	1) Push 2) Pop 3) Stack overflow 4) Stack underflow.	Each operation carries – 1 M
	e)	**What are directed and undirected graphs ?**	**2 M**
	Ans	**Directed Graph:** A directed graph G is also called **digraph** which is same as a multigraph except that each edge e in G is assigned a direction or in other words, each edge in G is identified with an ordered pair (U,V) of nodes in G rather than an unordered pair.	Each type carries – 1 M

		 G=(V,E) V(G)={V1,V2,V3,V4} E(g)={(V1,V2),(V1,V4),(V2,V4), (V1,V3)} **Undirected Graph:** An undirected graph G is a graph in which each edge e is not assigned a direction. 	
	f)	**Explain linear and non-linear data structures.**	**2 M**
	Ans	A linear data structure is one in which the components are stored in a **sequential order and are linked** to the elements before and after them. Array, Stack, Queue, Linked list, and other linear data structures are examples. When the data items or pieces of a data structure are not placed sequentially or linearly, the data structure is called to be non-linear. Because the items are not stored sequentially, they cannot be traversed or retrieved in a single iteration that is the single level is not engaged in non-linear data structures. In **terms of implementation**, a **non-linear data** structure is difficult. When compared to linear data structures, they make better use of system memory. Tree and graph are examples of non-linear data structures. The tree data structure contains a hierarchical relationship. Non-linear data structures are memory efficient because they do not require a memory allocation in advance, as previously stated.	Explanation carries – 1 M
	g)	**Define Searching. What are its types?**	**2 M**
	Ans	Searching is an operation or a technique that helps finds the place of a given element or value in the list. Any search is said to be successful or unsuccessful depending upon whether the element that is being searched is found or not. Some of the standard searching technique that is being followed in the data structure is listed below: • Linear Search or Sequential Search • Binary Search	Correct def: 1m, types- 1m
2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Sort the following elements using Radix Sort Method:**	**4 M**

{361, 12, 527, 143, 9, 768, 3481}.

Ans

First Iteration: Sort the numbers according to Unit place

Inputs	**Pockets**									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0361		0361								
0012			0012							
0527								0527		
0143				0143						
0009										0009
0768									0768	
3481		3481								

Output = {0361,3481,0012,0143,0527,0768,0009}

Second Iteration: Sort the numbers according to 10th place

Inputs	**Pockets**									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0361							0361			
3481									3481	
0012		0012								
0143					0143					
0527			0527							
0768							0768			
0009	0009									

Output = {0009,0012,0527,0143,0361,0768,3481}

Third Iteration: Sort the numbers according to 100th place

Inputs	**Pockets**									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0009	0009									

Each iteration carries – 1 M

0012	0012									
0527						0527				
0143		0143								
0361				0361						
0768								0768		
3481					3481					

Output = {0009,0012,0143,0361,3481,0527,0768}

Fourth Iteration: Sort the numbers according to 1000th place

Inputs	**Pockets**									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0009	0009									
0012	0012									
0143	0143									
0361	0361									
3481				3481						
0527	0527									
0768	0768									

Output = {0009,0012,0143,0361,0527,0768,3481}

Sorted Numbers are = 0009,0012,0143,0361,0527,0768,3481

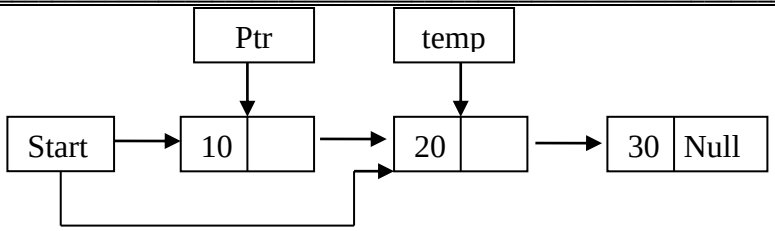
b) Write an algorithm to delete a node at the beginning from a singly Linked List. **4 M**

Ans

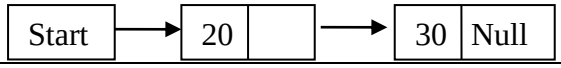
Deletefirst(start)

1. [check for Underflow]
   if start=NULL then print 'List is Empty'
   exit
   endif
2. set Ptr=start
3. set start=start->next
4. print 'Element deleted is Ptr->info
5. free(Ptr)

Correct logic carries – 4 M



After Deletion

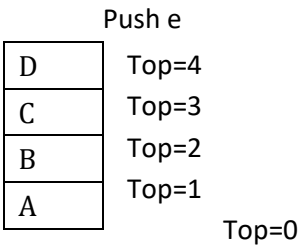


c) Explain stack overflow and underflow conditions with example. **4 M**

Ans

Stack Overflow condition: When stack is completely full (i.e. TOP= MaxSize -1) and we try to insert more element onto stack then this condition is called overflow condition and no further element could be inserted now until any element is deleted.

Stack Overflow: This is the situation when the stack becomes full, and no more elements can be pushed onto the stack. At this point the stack top is present at the highest location of the stack.



Underflow Condition: When a stack is empty (i.e. TOP= -1) and we try to delete more element from it, then this condition is called underflow condition.

Stack empty or underflow: This is the situation when the stack contains no element. At this point, the top of stack is present at the bottom of the stack.

stack overflow and underflow carries 2 M with example

d) Implement a C program to insert an element in an array. **4 M**

Ans

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

Any correct suitable program

```
int x,i,max=10,pos,K,n,a[10];
clrscr();
printf("Enter number of element");
scanf("%d",&n);
if(n<max)
{
printf("Enter the element:\n");
for(i=0;i<n;i++)
{
printf("Enter element %d\t",i+1);
scanf("%d",&a[i]);
}
printf("Array");
for(i=0;i<n;i++)
{
printf("\n element no %d is %d",i+1,a[i]);
}
printf("\n Enter the element to be added:");
scanf("%d",&x);
printf("Enter the postion of the element where element to be added");
scanf("%d",&pos);
for(i=n;i>=pos;i--)
{
a[i]=a[i-1];
}
a[pos-1]=x;
printf("Array with element inserted:");
for(i=0;i<n+1;i++)
{
printf("\n Element no %d is %d",i+1,a[i]);
}
}
else
{
printf("Memory not available....\n try again 1=y,2=n");
scanf("%d",&K);
if(K==1)
main();
else
exit();
}
getch();
}

//OUTPUT
/*Enter number of element5
Enter the element:
Enter element 1 23
Enter element 2 32
```

4 M

```
Enter element 3 65
Enter element 4 12
Enter element 5 45
Array
 element no 1 is 23
 element no 2 is 32
 element no 3 is 65
 element no 4 is 12
 element no 5 is 45
 Enter the position of element to be added:89
Enter the postion of the element where element to be added 1
Array with element inserted:
 Element no 1 is 89
 Element no 2 is 23
 Element no 3 is 32
 Element no 4 is 65
 Element no 5 is 12
 Element no 6 is 45
*/
```

3. **Attempt any THREE of the following:** **12 M**

a) **Differentiate between tree and graph with respect to any four parameters.** **4 M**

Ans

Parameter	Tree	Graph
Path	Only one between two vertices	More than one path allowed
Root node	It has exactly one root node	Graph may or may not have root node
Loop	No loop are permitted	Graph can have a loop
No. of edges	N-1	Not defined
complexity	Less complex as compared to graph	More complex as compared to tree
Traversal technique	Preorder, postorder and inorder	Breadth first search and depth first search
Parent child relationship	Parent child relationship exists	No parent child relationship exists
Application	Decision tree, sorting	Finding shortest path, route optimization.

Any 4 correct points – 4 M

b) **Write an algorithm to delete an intermediate node in a singly linked list.** **4 M**

Ans Step 1: IF START = NULL

Correct algorithm – 4 M

Write UNDERFLOW

Go to Step 1 [END OF IF]

Step 2: SET PTR = START

Step 3: SET PREPTR = PTR

Step 4: Repeat Steps 5 and 6 while PREPTR DATA != NUM

Step 5: SET PREPTR = PTR

Step 6: SET PTR = PTR NEXT [END OF LOOP]

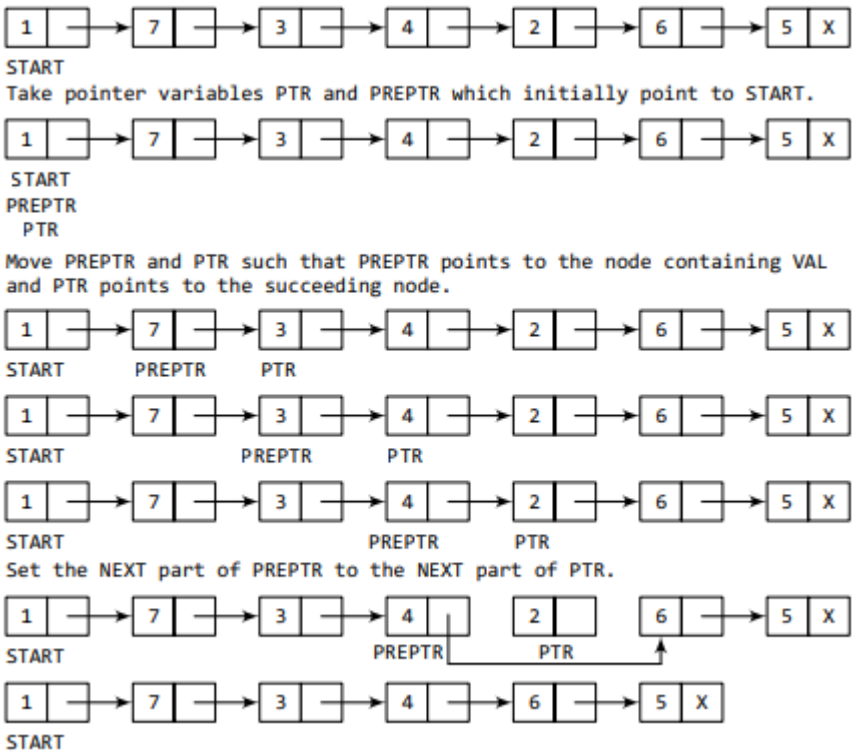
Step 7: SET TEMP = PTR

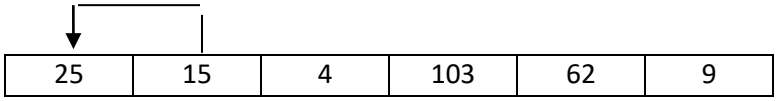
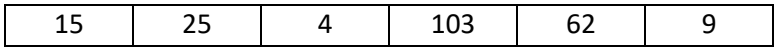
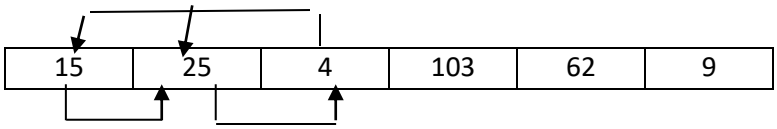
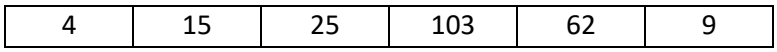
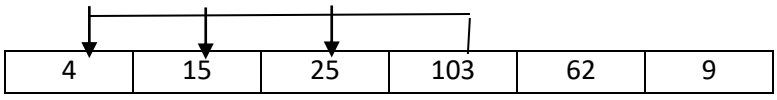
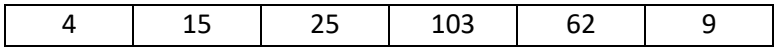
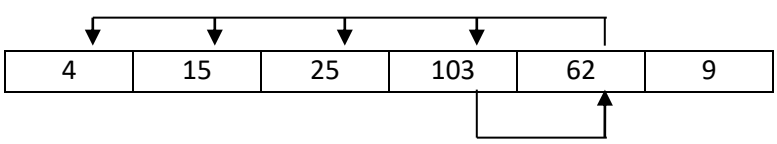
Step 8: SET PREPTR NEXT = PTR NEXT

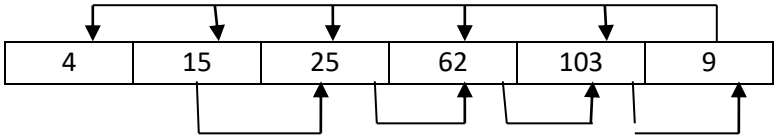
Step 9: FREE TEMP

Step 10 : EXIT

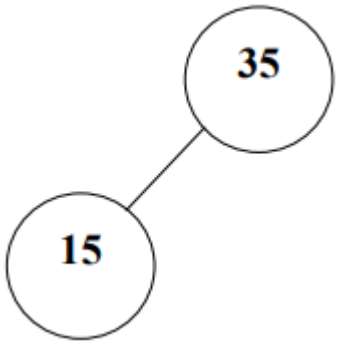
Consider the linked list shown in Fig. Suppose we want to delete the node that succeeds the node which contains data value 4. Then the following changes will be done in the linked list.



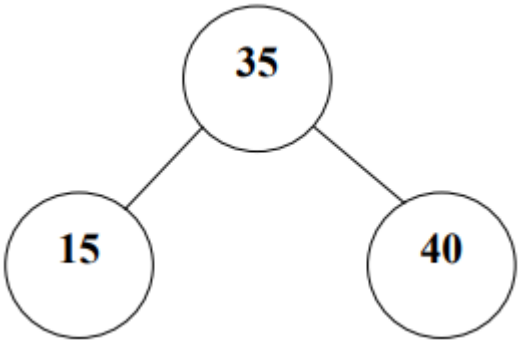
	c)	**Sort the following numbers in ascending order using Insertion sort:** **{25, 15, 4, 103, 62, 9} and write the output after each iteration.**	**4 M**
	Ans	Pass1: 1st element is compared with all previous element (i.e., 0th element)  Since 25>15, move 25 to right and insert (place) 15 to its correct position. Array elements after pass 1  Pass 2: Second element is compared with all previous element (i.e., 0th and 1st)  Since 15>4 and 25>4 move 15 and 25 to right by one position and insert (place) 4 to its correct position. Array element after pass 2  Pass 3: 3rd element is compared with all previous elements (i.e., 0th, 1st and 2nd)  Since 4<103, 15<103 and 25<103, 103 is at its correct position. Array element after pass 3  Pass 4: 4th element is compared with all its previous elements(i.e. 0th, 1st , 2nd and 3rd)  	Any Correct answer – 4 M

		<table><tr><td>4</td><td>15</td><td>25</td><td>103</td><td>62</td><td>9</td></tr></table> Compare 62 with 4, Since 62> 4, Compare 62 with 15, since 62> 15, Compare 62 with 25, since 62> 25. Again compare 62 with 103, since 62<103, move 103 to its right position by 1 element and insert (place) 62 to its right position i. e. a[3]. Array after pass 4 <table><tr><td>4</td><td>15</td><td>25</td><td>62</td><td>103</td><td>9</td></tr></table> Pass 5: 5th element is compared with all its previous elements(i.e. 0th, 1st , 2nd, 3rd and 4th)  Compare 9 with 4, Since 9> 4, compare 9 with 15, since 9< 15, Compare 9 with 25, since 9< 25. Compare 9 with 103, since 9< 103. Shift or move 15, 25, 62 and 103 to its right by 1 position and insert(place) 9 to its correct position i. e. a[1].	
	d)	**Construct the Binary Search Tree using following elements :** **(35, 15, 40, 7, 10, 100, 28, 82, 53, 25, 3}. Show diagrammatically each Step of construction of BST.**	**4 M**
	Ans	Step 1:- Inserting Element 35 	Correct answer – 4 M

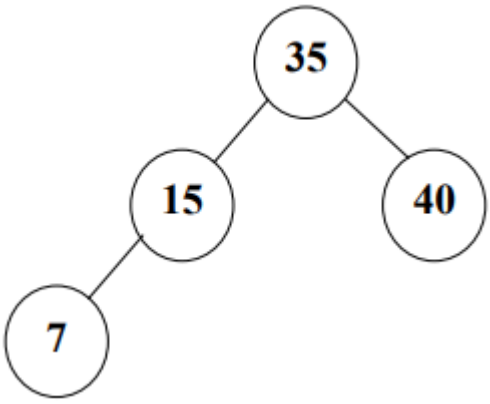
Step 2:- Inserting Element 15



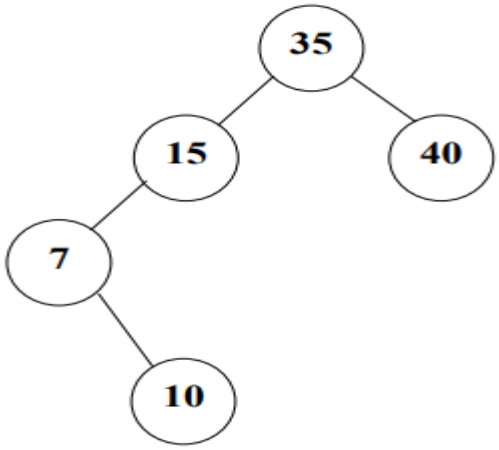
Step 3:- Inserting Element 40



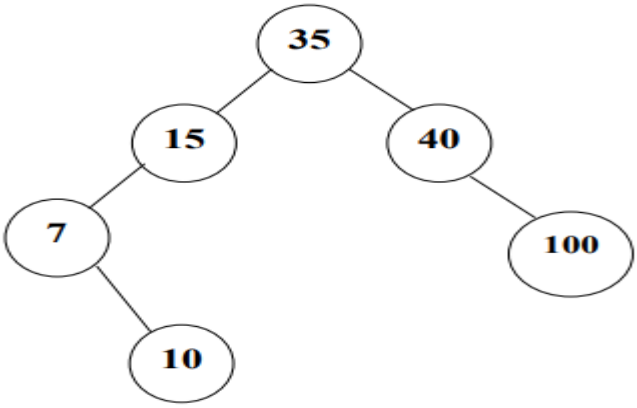
Step 4:- Inserting Element 7



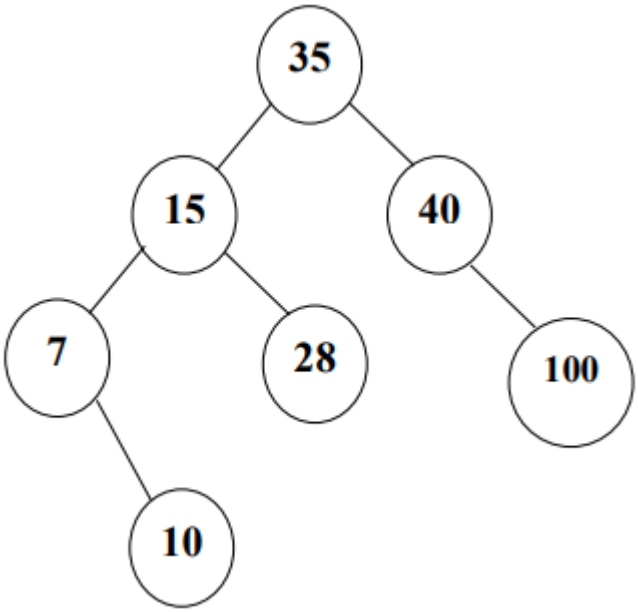
Step 5:- Inserting Element 10



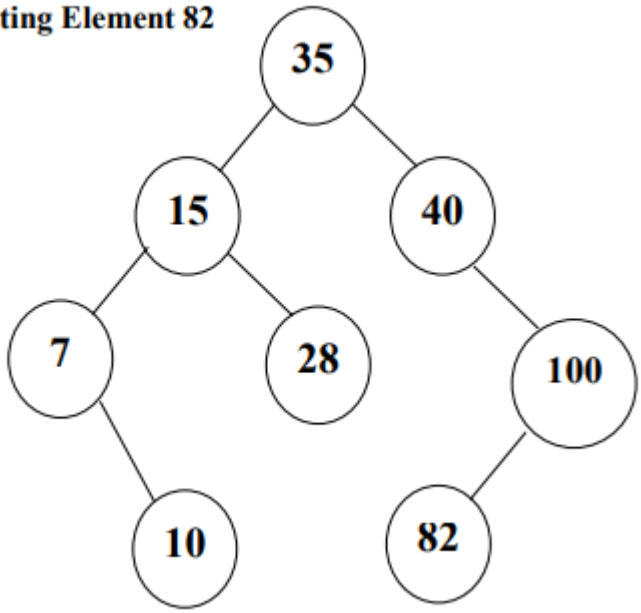
Step 6:- Inserting Element 100



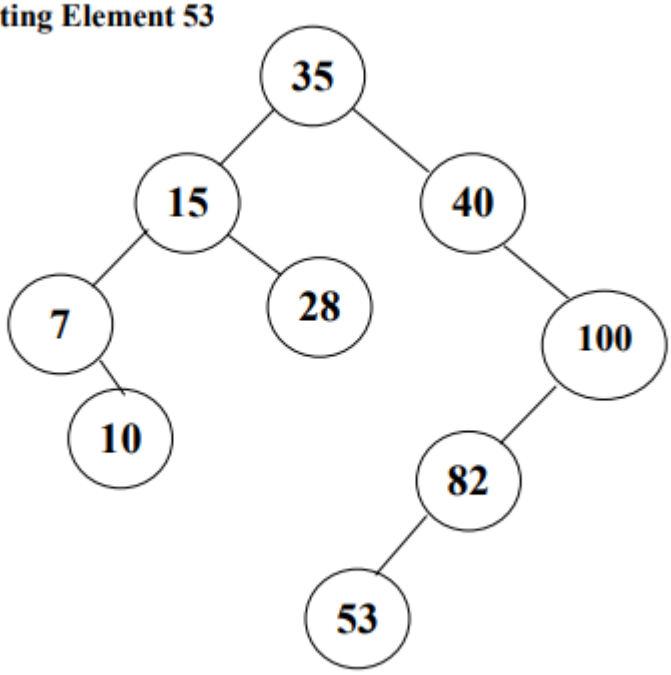
Step 7: - Inserting Element 28



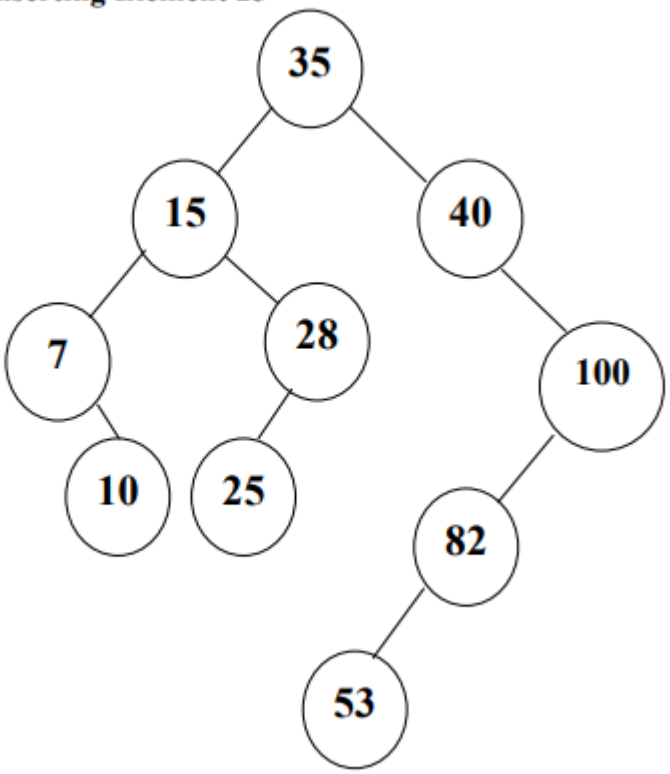
Step 8: - Inserting Element 82



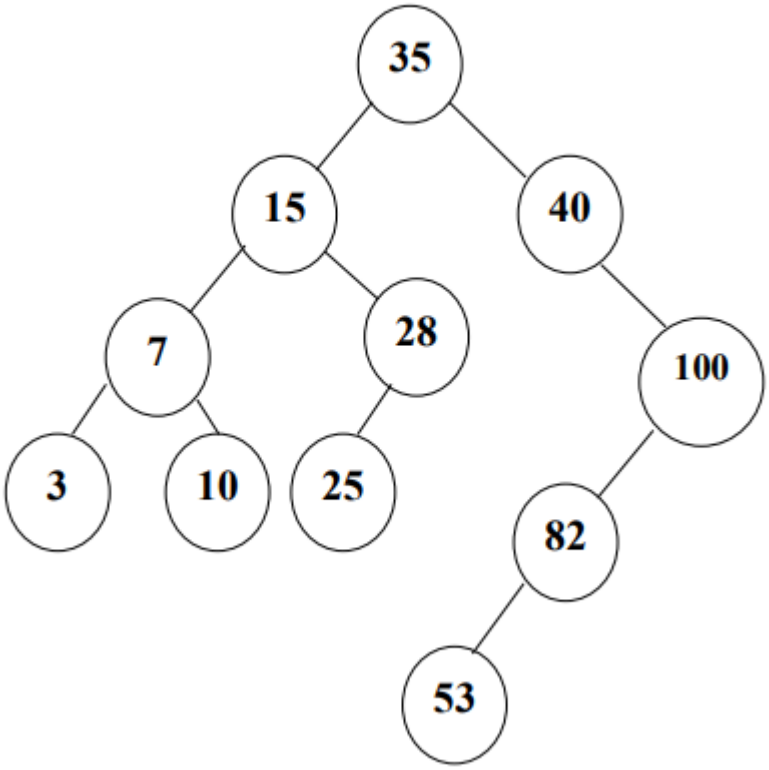
Step 9: - Inserting Element 53



Step 10:- Inserting Element 25

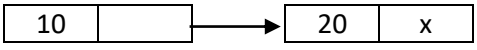


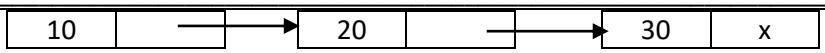
Step 11: - Inserting Element 3



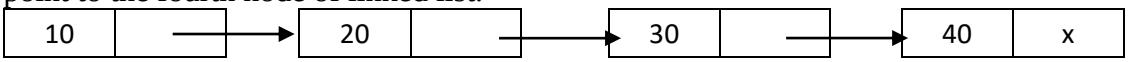
4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Differentiate between Binary search and Linear search with respect to any four parameters.**	**4 M**
	Ans		Any 4 correct points – 4 M

Linear Search	**Binary Search**
Search element is compared with each element in list	Search element is compared with mid element only
Simplest method of searching	Comparatively difficult method of searching
Easy to implement	Comparatively difficult to implement
Given list of numbers can be sorted or unsorted order	Given list of numbers must be in sorted order
Linear search only requires equality Comparisons.	Binary search requires an ordering comparison.
Linear search has complexity O(n).	Binary search has complexity O(log n).
Linear search is too slow to be used with large lists due to its o (n) average case performance.	Binary search is considered to be a more efficient method that could be used with large lists.
Linear search only requires sequential Access.	Binary search requires random access to the data.

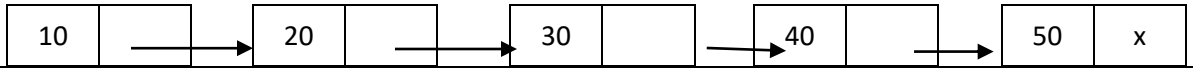
	b)	**Create a singly Linked List using data fields 10, 20, 30, 40, 50 and show procedure step-by-step with the help of diagram from start to end.**	**4 M**
	Ans	Step 1: For creating a singly linked list, first create new node (first node) allocate memory for new node and initialised its data part to 10 and next part to null. Start address will point to the first node of linked list. 10 | X Start Step 2: To add 20 in singly linked list, create new node (second node) allocate memory for new node and initialised its data part to 20 and next part to null. Address of first node will point to the second node of linked list.  Step 3: To add 30 in singly linked list, create new node (third node) allocate memory for new node and initialised its data part to 30 and next part to null. Address of second node will point to the third node of linked list.	Correct answer – 4 M



Step 4: To add 40 in singly linked list, create new node (fourth node) allocate memory for new node and initialised its data part to 40 and next part to null. Address of third node will point to the fourth node of linked list.



Step 5: To add 50 in singly linked list, create new node (fifth node) allocate memory for new node and initialised its data part to 50 and next part to null. Address of fourth node will point to the fifth node of linked list.



c) **Show the effect of PUSH and POP operation on to the stack of size 10. The stack contains 10, 20, 30, 40, 50 and 60, with 60 being at top of the stack. Show diagrammatically the effect of -** **4 M**

(i) PUSH 55

(ii) PUSH 70

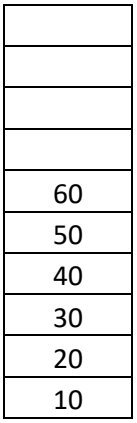
(iii) POP

(iv) POP

Sketch the final structure of stack after performing the above said operations.

Ans

Initial Stack



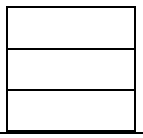
1 M for push 55

1 M for push 70

1 M for pop

1 M for pop

(i) PUSH 55



55	top =6
60	
50	
40	
30	
20	
10	

(ii) PUSH 70

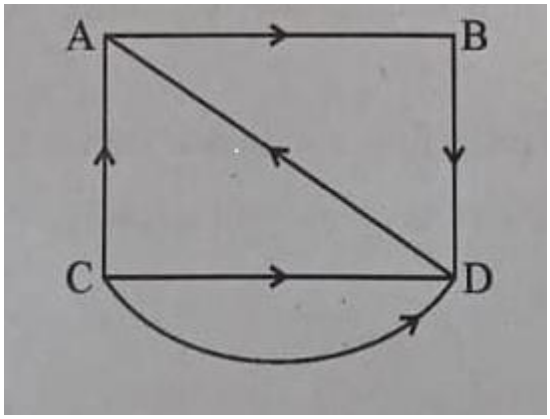
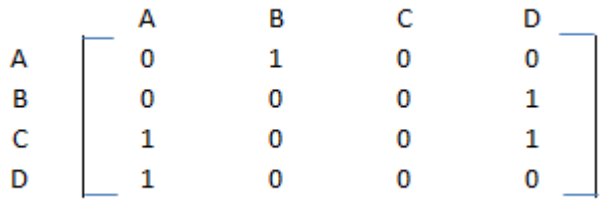
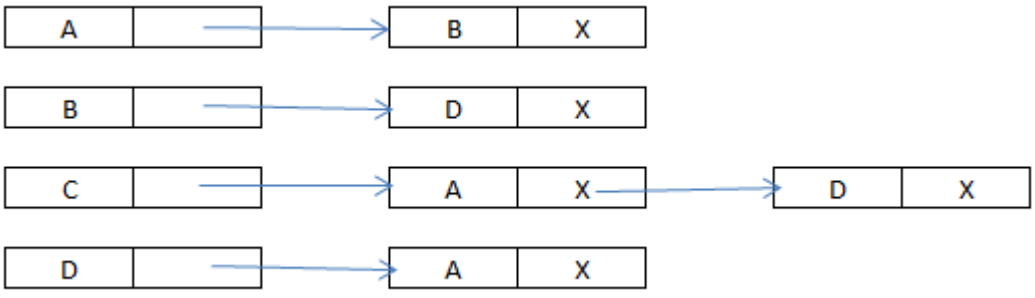
70	top =7
55	
60	
50	
40	
30	
20	
10	

(iii) POP

55	top =6
60	
50	
40	
30	
20	
10	

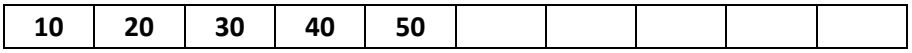
(iv) POP

60	top =5

		50 40 30 20 10	
	d)	**For the following directed graph:** **(i) Give adjacency matrix representation.** **(ii) Give adjacency list representation.** 	**4 M**
	Ans	Adjacency matrix representation  Adjacency list representation 	Matrix representation – 2 M, List representation – 2 M

5.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Convert the infix expression to its postfix expression using stack ((A+B)*D)^(E-F).Show diagrammatically each step of conversion.**	**6 M**
	Ans		Correct postfix expression- 6 M (Note: Correct Steps should be considered and given ½ M)

Input Symbol	Stack	Postfix Expression
	#	
(	# (	EMPTY
(	# ((	EMPTY
A	# ((	A
+	# ((+	A
B	# ((+	AB
)	# (	AB+
*	# (*	AB+
D	# (*	AB + D
)	#	AB + D*
^	# ^	AB + D*
(	# ^ (	AB + D*
E	# ^ (	AB + D *E
-	# ^ (-	AB + D *E
F	# ^ (-	AB + D *E F
)	# ^	AB + D *E F -
	Stack empty	AB + D *E F - ^

	b)	**Show the effect of INSERT and DELETE operation onto the linear queue of size 10. The linear queue sequential contains 10,20,30,40 and 50 where 10 is at front queue. Show diagrammatically the effect of –** **INSERT (75)** **INSERT (85)** **DELETE** **INSERT (60)** **DELETE** **INSERT (90)**	**6 M**
	Ans	Given Queue  front rear Insert (75)	Each correct operation – 1 M

10	20	30	40	50	75				

front rear

Insert (85)

10	20	30	40	50	75	85			

front rear

Delete

	20	30	40	50	75	85			

front rear

10 is deleted

Insert (60)

	20	30	40	50	75	85	60		

front rear

Delete

		30	40	50	75	85	60		

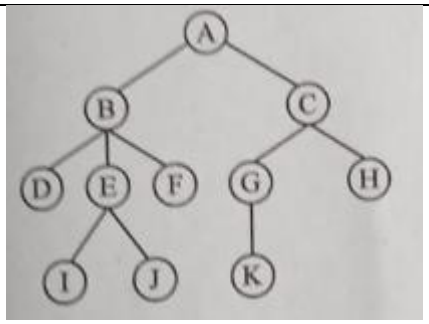
front rear

20 is deleted

Insert (90)

		30	40	50	75	85	60	90	

front rear

	c)		**6 M**

From the given tree, complete the following answers :

(i) Degree of tree : _____
(ii) Degree of node B: _____
(iii)Level of node H: _____
(iv)Indegree of node C: _____
(v) (V) Outdegree of node B : _____
(vi)(vi) Height of the tree : _____

	Ans	(i) Degree of tree: 3 (ii) Degree of node B: 2 (iii)Level of node H: 2	Each correct answer –

		(iv) Indegree of node C: 1 (v) Outdegree of node B: 3 (vi) Height of the tree: 3	1 M
6.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Find the position of element 29 using Binary search method in an array given as : { 11,5,21,3,29,17,2,43}**	**6 M**
	Ans	**Given Array :** **0 1 2 3 4 5 6 7** a | 11 | 5 | 21 | 3 | 29 | 17 | 2 | 43 | sorted array **0 1 2 3 4 5 6 7** a | 2 | 3 | 5 | 11 | 17 | 21 | 29 | 43 | Pass-1 **0 1 2 3 4 5 6 7** a | 2 | 3 | 5 | 11 | 17 | 21 | 29 | 43 | L H mid= (L+H)/2 = (0+7)/2 = 7/2=3 **0 1 2 3 4 5 6 7** a | 2 | 3 | 5 | 11 | 17 | 21 | 29 | 43 | L mid H a[mid]=a[3]=11 11 is not equal to 29 Since 29 > 11 L = mid+1 = 4 H = n-1=7 **Pass-2** **4 5 6 7** a | 17 | 21 | 29 | 43 | L H mid = 5 **4 5 6 7** a | 17 | 21 | 29 | 43 | L mid H	Each correct pass-2M

a[mid]=a[5]=21
21 is not equal to 29
Since 29 > 21
L = mid+1 = 6
H = n-1=7

Pass- 3

	6	**7**
a	29	43
	L	H

mid = 6
a[mid]=a[6]=29
29 is equal to 29

Hence Element 29 Is found at position 7th of array

b) Evaluate the following postfix expression : **6 M**

4 6 24 + *6 3 / -

Show diagrammatically each step of evaluation using stack.

Ans

postfix expression	**operand 1**	**operand 2**	**result**
4			4
6			4 6
24			4 6 24
+	6	24	4 30
*	4	30	120
6			120 6
3			120 6 3
/	6	3	120 2
-	120	2	118

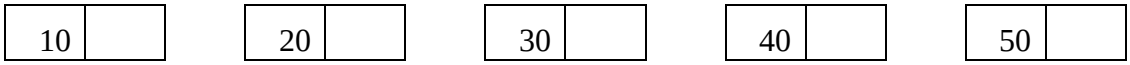
Each correct steps- 1 M

Result is 118

c) Create a singly linked list using data fields 10, 20, 30, 40, 50. Search a node 40 from the singly linked list and show procedure step-by-step with the help of the diagram from start to end. **6 M**

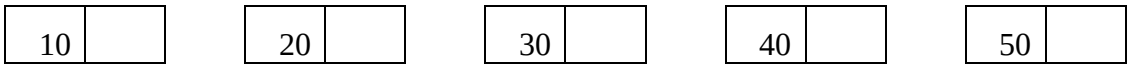
Ans

For correct answer 6m



Node to be searched = 40

Consider pointer ptr is used for traversal of Singly linked List

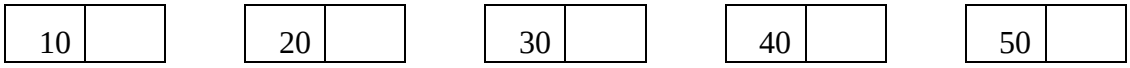


ptr

will check ptr->infor =10

Since 10 is not equal to 40

set ptr = ptr->next

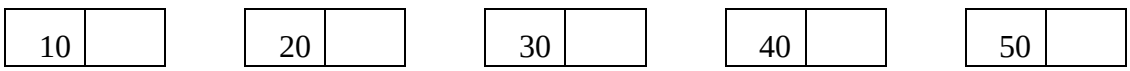


ptr

will check ptr->infor = 20

Since 20 is not equal to 40

set ptr = ptr->next

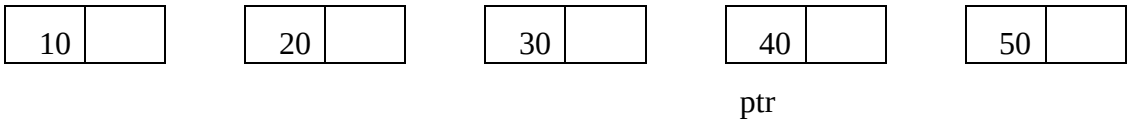


ptr

will check ptr->infor = 30

Since 30 is not equal to 40

set ptr = ptr->next



will check ptr->infor = 40

Since 40 is equal to 40

Hence Node 40 is present in Linked list