PRELAB QUESTIONS-3:

1) Write an algorithm of the tree by using preorder traversal without recursion.

```
Algorithm IterativePreorder(node *root)
{ if (root ==NULL)
    return;
  stack node * nodeStack;
  nodeStack.push(root);
  while (nodeStack.empty() == false)
  { struct node *node = nodeStack.top();
    printf("%d", node->data);
    nodeStack.pop();
    if(node->right)
      nodeStack.push(node->right);
    if(node->left)
      nodeStack.push(node->left);}
}
```

VASAVI COLLEGE OF ENGINEERING

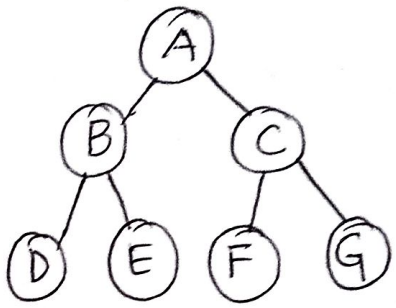
(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA LAB

Name K.S.I.SIVANI Roll No. D52 Page No. 32

2) Write tree traversal of the tree given below in 3 methods:



Inorder: LNR

DBEAFCG

Preorder: NLR

ABDECFG

Postorder: LRN

DEBFGCA

3) What is the time complexity of post order traversal using iterative approach?

```
void postOrderIterative(struct node *root)
{ if(root == NULL) return;
   struct Stack *stack = createStack(MAX_SIZE);
   do
   { while(root)
     { if(root->right)
          push(stack, root->right);
       Push(stack, root);
```

```
        root = root->left; }
        root = pop(stack);
        if (root->right && peek(stack) == root->right)
        { pop(stack);
          push(stack, root);
          root = root->right; }
        else
        { printf("%d", root->data);
          root = NULL; }
        While(!isEmpty(stack));
}
```

Time complexity = O(n)

```
void preorder(~~root~~ node *t)
{ if(!t) { printf("%d", t->data);
    preorder(t->lchild);
    preorder(t->rchild); }
}
```

4) Design an algorithm to print all even nodes of a BST. What is the time complexity.

```
Algorithm EvenNode (Node *root)
{   if (root!=NULL)
        evenNode (root->left);
    if (root->key %2==0)
        printf("%d ", root->key);
    evenNode(root->right);}
```

Time Complexity: O(N). N → no. of nodes.

5) Derive the time complexity of the algorithm to determine the height of a BST.

```
Algorithm findHeight (Node *node)
{ if (node == NULL)
      return 0;
  else
  { int leftHeight (node->left);
    int rightHeight (node
```

```
else
{ int leftHeight = findHeight (node→left);
  int rightHeight = findHeight (node→right);
  if (leftHeight > rightHeight)
        return (leftHeight +1);
  else
        return (rightHeight+1); } }
```

Time Complexity : O(n).

6) Which traversal of a BST gives the sorted order of the elements? Design the recursive algorithm for it and write the time complexity.

→ Inorder traversal gives the sorted order of elements in binary search tree.

```
Algorithm inorder (node *t)
{  if (!t)
   { inorder (t→lchild);
     printf("%d",t→data);
     inorder (t→rchild); }
}
```

Time Complexity = O(n).

7) Design an algorithm for level order traversal of a complete BST and determine its time complexity.

```
Algorithm printLevelOrder (struct node *root)
{ int h:=height(root);
  int i;
  for& i:= 1 to h
        printCurrentLevel (root, i); }

Algorithm printCurrentLevel (struct node *root, int level)
{ if (root == NULL)
      return;
  if (level == 1)
      printf("%d", root->data);
  else if (level > 1)
  { printCurrentLevel (root->left, level-1);
    printCurrentLevel (root->right, level-1); }
}
```

* ~~LAB PROGRAMS-3~~

* PRELAB PROGRAMS-3:

1) Implement program to construct full binary tree using its preorder and preorder traversal of its mirror tree.

```
#include <stdio.h>
#include <stdlib.h>
struct node
{ int item;
  struct node *left;
  struct node *right;};
void preorderTraversal(struct node *root)
{ if (root==NULL) return ;
  printf("%d→ ", root→item);
  preorderTraversal(root→left);
  preorderTraversal(root→right);}
struct node *createNode(int value)
{struct node *newNode=malloc(sizeof(struct node));
  newNode→item=value;
  newNode→left=NULL;
```

```
    newNode->right = NULL;
    return newNode; }
struct node * insert(struct node * root, int x)
 { if(root == NULL)
      return createNode(x);
   else if (x > root->item)
      root->right = insert(root->right, x);
   else
      root->left = insert(root->left, x);
   return root;
void mirror(struct node * node)
 { if (node == NULL)
      return ;
   else
     { struct node * temp;
       mirror(node->left);
       mirror(node->right);
       temp = node->left;
       node->left = node->right;
       node->right = temp; } }
```

Output:

```
Enter the node element: 5
Enter the no of nodes:6
1
2
3
4
6
7
```

Preorder traversal

5 → 1 → 2 → 3 → 4 → 6 → 7 →

Preorder traversal of Mirror:

5 → 6 → 7 → 1 → 2 → 3 → 4 →

COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA LAB

Name K.S.I.SIVANI Roll No. -052 Page No. 39

```
int main()
{ int m,n,i,j,x;
  printf("Enter the node element: ");
   scanf("%d", &m);
    struct node *root = createNode(m);
    printf("Enter the no. of nodes:");
     scanf("%d", &n);
     for(i=0; i<n; i++)
     { scanf("%d",&x);
       insert(root, x); }
     printf("\n Preorder Traversal\n");
     preorderTraversal(root);
     mirror(root);
     printf("\n Preorder Traversal of mirror:\n");
     preorderTraversal(root);
     return 0; }
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA LAB

Name K.S.I. SIVANI Roll No. 1602-21-733-052 Page No. 40

2) Implement program to find an inorder successor & predecessor of specified node in a tree.

```
#include <stdio.h>
#include <stdlib.h>
struct node
{ int data;
  struct node * left;
  struct node * right; };
struct node *creatNode(int data)
{ struct node *n;
   n=(struct node *) malloc (sizeof(struct node));
   n->data=data;
   n-> left =NULL;
   n-> right=NULL;
   return n; }
void inorder (struct node *root)
{ if(root ==NULL) return;
  inorder (root-> left);
  printf("%d \t", root->data);
```

```
inorder(root->right);}
struct node *predec(struct node *node)
 { struct node *current=node;
   if(current && current->left!=NULL)
      current = current -> left;
   return current;}

 struct node *succes(struct node *node)
 { struct node *current=node;
    if(current && current->right!=NULL)
       current=current->right;
    return current;}
int main()
{ struct node *p = createNode(5);
  struct node *root=p;
   struct node *p1 =createNode(3);
   struct node *p2 = createNode(6);
   struct node *p3 = createNode(1);
   struct node *p4 =createNode(4);
```

* Output:

1 3 4 5 6
1
6


```
p->left=p1; p->right=p2;
p1->left=p3; p1->right=p4;
inorder(root);
struct node *temp = predec(p1);
struct node *temp1 = succes(p);
printf("\n%d", temp->data);
printf("\n%d", temp1->data);
return 0;}
```

3) Implement post order traversal of binary tree without recursion and stack:

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
struct node
{int data;
  struct node *left;
  struct node *right;
  int visited;
};
```

```
Struct node *createNode(int data)
{ struct node *n;
  n=(struct node *) malloc (sizeof(struct node));
  n->data = data;
  n->left = NULL;
  n->right = NULL;
  n->visited =0;
  return n; }
void postorder (struct node *root)
{ struct node *temp = root;
  While ((temp != NULL) && (temp->visited)==0)
  { if(temp->left!=NULL && (temp->left)->visited
                                    ==0)
      temp = temp->left;
    else if (temp->right != NULL && (temp->right)->visited
                                    ==0)
      temp=temp->right;
    else
      { printf("%d ", temp->data);
        temp->visited=1;
        temp = root; } } }
```

O/P:

Start 1 4 3 6 5.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hydorabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DBMS

Name K.S.T.SIVANI Roll No. -052 Page No. 44

```
int main()
{ struct node *p =createNode(5);
  struct node *p1=createNode(3);
  struct node * p2 =createNode(6);
  struct node *p3= createNode(1);
  struct node * p4 = createNode(4);
  p→left = p1;
  p→right = p2;
  p1→ left = p3;
  p1→ right = p4;
  printf("start");
  postorder(p);
  return 0; }
```

LAB PROGRAMS-3

1) WAP to implement randomized quick sort:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
void quicksort(int x[20], int p, int q)
{ int temp, i, j, pivot;
  if(p<q)
  { i=p; j=q;
    pivot = p+rand()%(q-p);
    temp=x[pivot];
    x[pivot]=x[p];
    x[p]=temp;
   while(i<j)
   { while(x[i]<=x[pivot] && i<q) { i++; }
     while (x[j]>x[pivot] {j--;}
     if(i<j)
     { temp=x[i]; x[i]=x[j]; x[j]=temp;}}
```

O/p:

Enter the no. of elements: 6

5
9
7
1
2
3

Sorted: 2 3 4 5 7 9.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : DAA

Name K.S.I.SIVANI Roll No. -052 Page No. 46

```
    temp=x[pivot]; x[pivot] = x[j]; x[j]=temp;
    quicksort(x,p,j-1);
    quicksort(x,j+1,q);}
}
int main()
{ int n;
  printf("Enter the no.of elements:");
  scanf("%d",&n);
   int a[n];
   int i;
   for(i=0; i<n; i++)
   {scanf("%d",&a[i]);}
    quicksort(a,0,n-1);
    printf("Sorted:");
    for(i=0; i<n; i++)
     {printf("%d\t",a[i]);}
     return 0;
   }
```