LAB - 8 15/02/24

1) Create a binary search tree perform pre-order post-order and in-order and display contents.

```
#include <stdio.h>
#include <stdlib.h>

struct node
{
    int data;
    struct node * left;
    struct node * right;
};
struct node * root = NULL;

void create()
{ struct node* new node = (struct node*) malloc (sizeof(struct node));
  printf("Enter the data to be inserted \n");
  scanf("%d", &new node -> data);
  new node -> left = NULL; new_node -> right = NULL;
  if (root == NULL)
  {
      new_node -> left = NULL;
      new_node -> right = NULL;
      root = new_node;
  }
  else
  {
      ptr = root;
      while (ptr -> left != NULL || ptr -> right != NULL)
      {
          if (new_node -> data > ptr -> data)
          {
```

```
if (ptr -> right != NULL)
{
    ptr = ptr -> right;
}
else
{
    break;
}
else
{
    if (ptr -> left != NULL)
    {
        ptr = ptr -> left;
    }
    else
    {
        break;
    }
}
if (new_node -> data > ptr -> data)
{
    ptr -> right = new_node;
}
else
{
    ptr -> left = new_node;
}
}
```

```
void pre_order (struct node *ptr)
{
    struct node * trav = ptr;
    if (ptr != NULL)
    {
        printf ("%d", ptr -> data);
        pre_order (ptr -> left);
        pre_order (ptr -> right);
    }
}

void inorder (struct node * ptr)
{
    struct node * trav = ptr;
    if (ptr != NULL)
    {
        inorder (ptr -> left);
        printf ("%d", ptr -> data);
        inorder (ptr -> right);
    }
}

void post_order (struct node * ptr)
{
    struct node * trav = ptr;
    if (ptr != NULL)
    {
        post_order (ptr -> left);
        post_order (ptr -> right);
        printf ("%d", ptr -> data);
    }
}
```

```
void main()
{
    printf(" Enter\n 1. Create\n 2. Pre-order\n 3. In-order
        4. post-order\n 5. Exit\n");
    int ch;
    do {
        printf (" Enter your choice\n");
        scanf ("%d", &ch);
        Switch (ch)
        {
            case 1: create();
            break;
            case 2: pre-order (root);
            break;
            case 3: inorder (root);
            break;
            case 4: post-order (root);
            break;
        }
    } while (ch != 5);
}
```

output:

Enter
1. Create
2. Pre-order
3. In-order
4. post-order
5. EXIT

Enter your choice
1

Enter the data to be inserted
4
Enter the choice
1
Enter the data to be inserted
6
Enter your choice
1
Enter the data to be inserted
1
Enter the choice
1
Enter the data to be inserted
3
Enter the choice
1
Enter the data to be inserted
5
Enter your choice
3
13456 Enter your choice
2
4136 5 Enter your choice
4
31564 Enter your choice
5