15/1/24

Binary Search Tree

```
#include<stdio.h>
#include<stdlib.h>
struct node
 {
   int value;
   struct node *left;
   struct node *right;
 } *root = NULL , * temp = NULL , ~~*t2~~, ~~*t1~~, ~~*t~~p;
   root points to                  temporary pointers
     binary search
         tree
 ~~int key~~;
 ~~int s~~;
 ~~int n; x~~
 ~~int count;~~
 ~~void search(struct node *t);~~

 void insert()
 {
  int data;
  printf(" Enter data to be inserted - ");
  scanf(" %d", &data);
      temp = (struct node *) malloc (sizeof(struct node));
      temp->value=data;
      temp->left = temp->right = NULL;
  if(root == NULL)
      root = temp;
  else
      search (root);
 }
```

```
void search(struct node *t)
{
 if((temp->value > t->value) && (t->right != NULL))
   search(t->right);
 else if((temp->value > t->value) && (t->right == NULL))
   t->right = temp;
 else if((temp->value < t->value) && (t->left != NULL))
   search(t->left);
 else if((temp->value < t->value) && (t->left == NULL))
   t->left = temp;

}
void inorder(struct node *t)
{
 if(root == NULL)
  {
  printf("No elements in the tree\n");
  return;
  }
 if(t->left != NULL)
 inorder(t->left);
 printf("%d->", t->value);
 if(t->right != NULL)
 inorder(t->right);
 }
void preorder(struct node *t)
{
 if(root == NULL)
  {
  printf("No elements in the tree\n");
  return;
  }
 printf("%d->", t->value);
 if(t->left != NULL)
 preorder(t->left);
 if(t->right != NULL)
 preorder(t->right);
}
```

[illegible]
15/2/24.

```
void postorder(struct node *t)
{
    if(root == NULL)
    {
        printf("No elements in the tree \n");
        return;
    }
    if(t->left != NULL)
    postorder(t->left);
    if(t->right != NULL)
    postorder(t->right);
    printf("%d->", t->value);
}
struct node* maxvaluenode(struct node *t)
{
    struct node * current = t;
    while (current && current->right != NULL)
        current = current->right;
    return current;
}

int main()
{
    int choice;
    while (1)
    {
        printf("\n menu\n");
        printf("1. insert an element into tree \n");
        printf("2. to get the max value in the tree \n");
        printf("3. to print the tree elements in inorder traversal\n");
        printf("4. to print the tree elements in preorder traversal\n");
        printf("5. to print the tree elements in postorder traversal\n");
        printf("6. to exit\n");
        printf("Enter your choice\n");
        scanf("%d", &choice);
switch (choice)
{
case1:
    insert();
    break;

case 2:
    tp = maxvaluenode(root);
    printf(" Max value node ");
    printf(" Node with Maximum value = %d", tp->value);
    break;

case 3:
    printf(" inorder traversal \n");
    inorder( root );
    break;

Case 4:
    printf(" preorder traversal \n");
    preorder( root);
    break;

Case 5:
    printf(" Postorder traversal \n");
    postorder (root);
    break;

Case 6:
    exit(0);
default :
    printf(" Invalid choice");
    break;
  }
}
return 0;
}
```

ng

in,

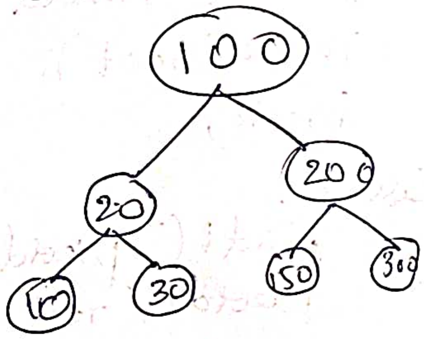
l\n");

sal\n")

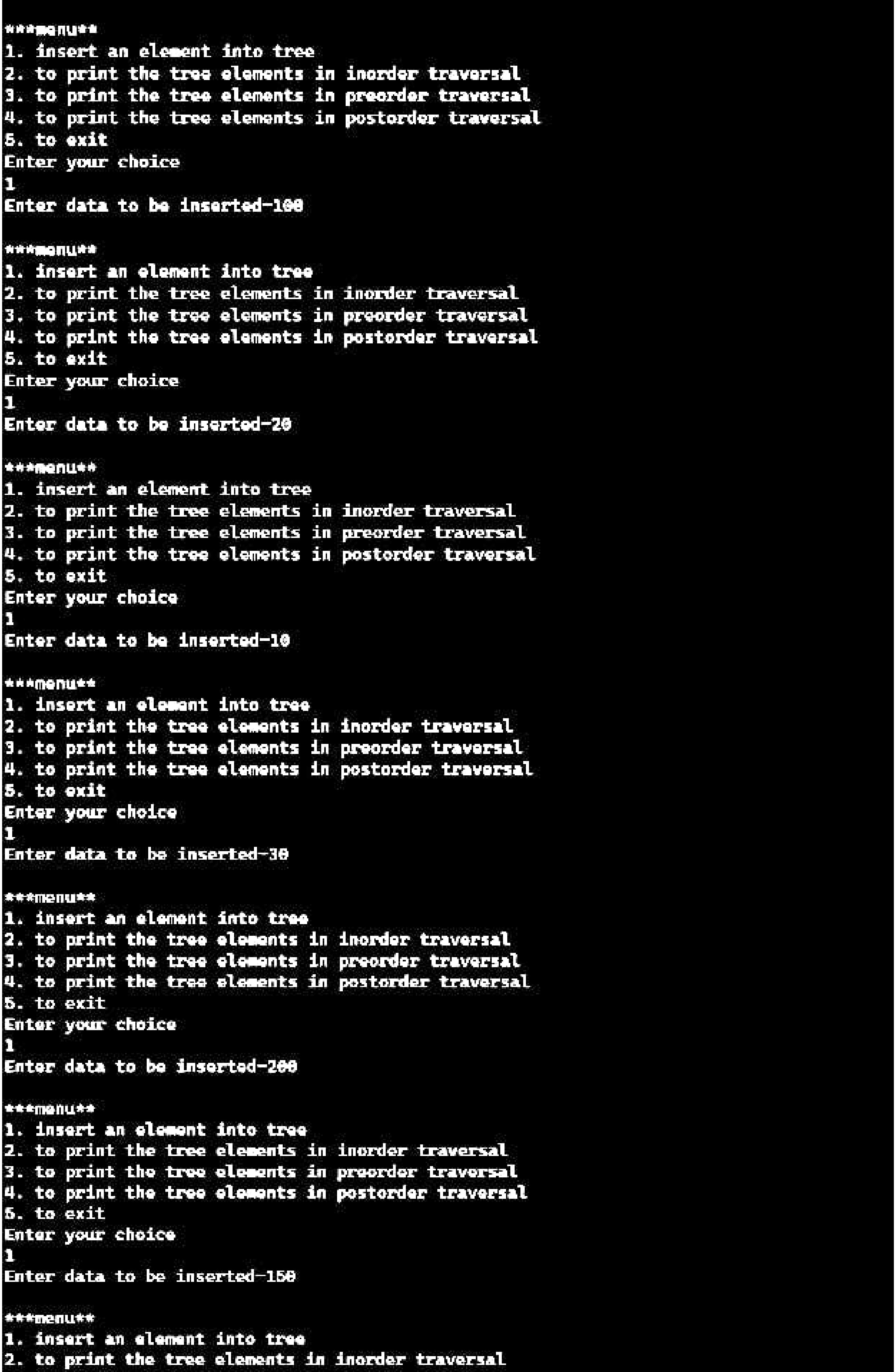
rsal\n")

o/p :-

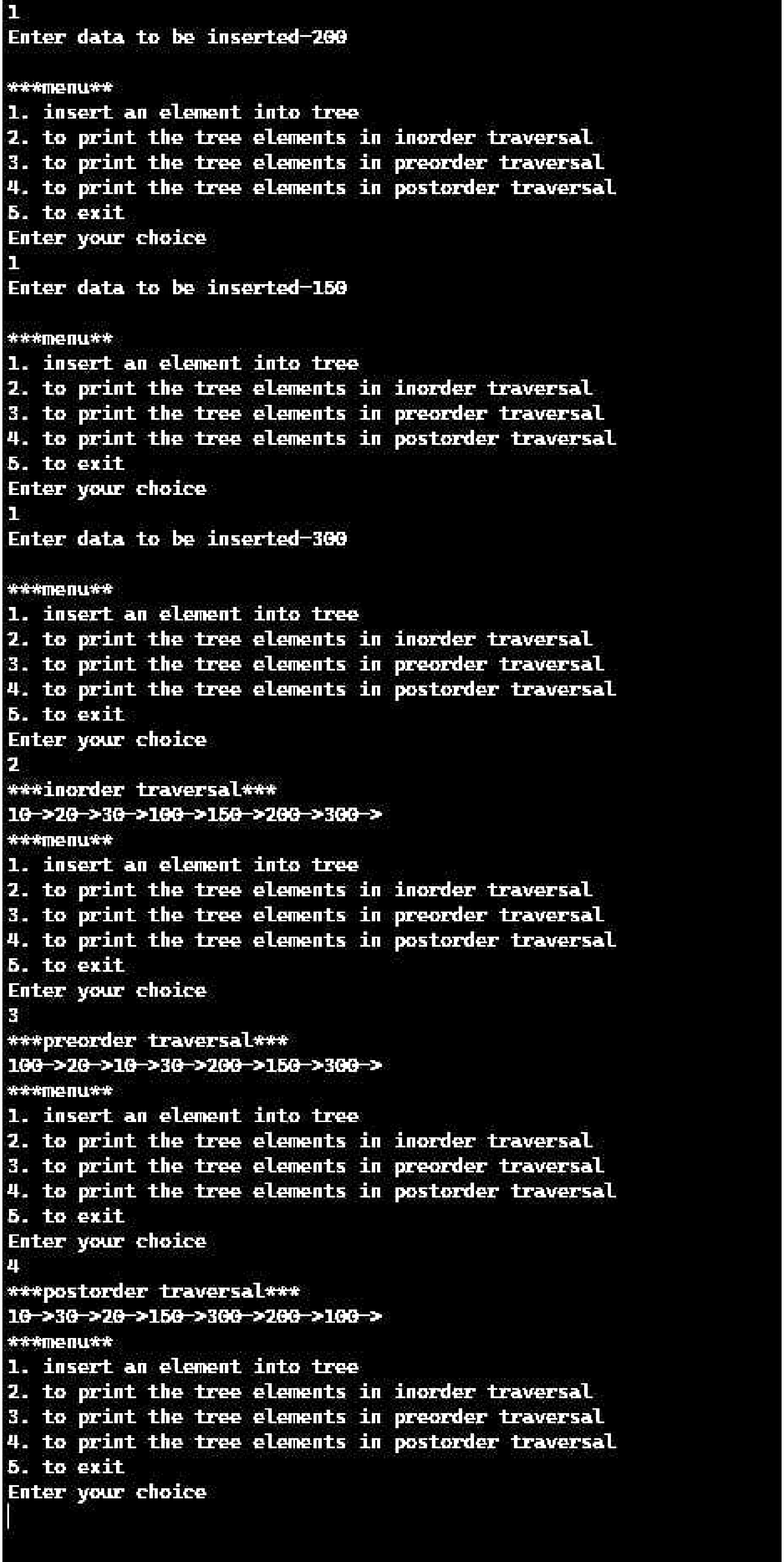
```
menu
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
> Enter your choice
1
Enter data to be inserted 100
20
10
30
200
150
300
> Enter your choice
2
                                        left, Root, Right
inorder traversal
   10 -> 20 -> 30 -> 100 -> 150 -> 200 -> 300 ->
                                        Root, left, Right
preorder traversal
   100 -> 20 -> 10 -> 30 -> 200 -> 150 -> 300 ->
                                        left, Right, root
postorder traversal
   10 -> 30 -> 20 -> 150 -> 300 -> 200 -> 100 ->
```



[illegible]
15/2/24



```
***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-100

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-20

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-10

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-30

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-200

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-150

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
```



```
1
Enter data to be inserted-200

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-150

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
1
Enter data to be inserted-300

***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
2
***inorder traversal***
10->20->30->100->150->200->300->
***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
3
***preorder traversal***
100->20->10->30->200->150->300->
***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
4
***postorder traversal***
10->30->20->150->300->200->100->
***menu**
1. insert an element into tree
2. to print the tree elements in inorder traversal
3. to print the tree elements in preorder traversal
4. to print the tree elements in postorder traversal
5. to exit
Enter your choice
```