BST

No1:

void levelOrderTraversal(BSTNode\* root)

{

    Queue q;

    q.head = NULL;

    q.tail = NULL;

    BSTNode \*temp;

    temp = root;

    if(temp == NULL) return;

    else{

        enqueue(&q.head, &q.tail, temp);

        while(!isEmpty(q.head)){

            temp = dequeue(&q.head, &q.tail);

            printf("%d",temp->item);

            if(temp->left != NULL){

                enqueue(&q.head, &q.tail, temp->left);

            }

            if(temp->right != NULL){

                enqueue(&q.head, &q.tail, temp->right);

            }

        }

    }

}

No2:

void inOrderTraversal(BSTNode \*root)

{

    /\* add your code here \*/

    Stack s;

    s.top = NULL;

    BSTNode \*temp;

    temp = root;

    if(temp == NULL){}

    else{

        while(1){

            if(temp != NULL){

                push(&s, temp);

                temp = temp->left;

            }

            else{

                if(!isEmpty(&s)){

                    temp = pop(&s);

                    printf("%d",temp->item);

                    temp = temp->right;

                }

                else{

                    break;

                }

            }

        }

    }

}

No3:

void preOrderIterative(BSTNode \*root)

{

    /\* add your code here \*/

    Stack s;

    s.top = NULL;

    BSTNode \*temp;

    temp = root;

    if(temp == NULL){}

    else{

        push(&s, temp);

        while(!isEmpty(&s)){

            temp = pop(&s);

            printf("%d ",temp->item);

            if(temp->right != NULL) push(&s, temp->right);

            if(temp->left != NULL) push(&s, temp->left);

        }

    }

}

No.4:

void postOrderIterativeS1(BSTNode \*root)

{

Stack s;

    BSTNode \*temp;

s.top = NULL;

temp = root;

if (temp == NULL)

return;

do

{

while (temp)

{

if (temp->right)

push(&s, temp->right);

push(&s, temp);

temp = temp->left;

}

temp = pop(&s);

if (temp->right && peek(&s) == temp->right)

{

pop(&s);

push(&s, temp);

temp = temp->right;

}

else

{

printf("%d ", temp->item);

temp = NULL;

}

} while (!isEmpty(&s));

}

No.5

void postOrderIterativeS2(BSTNode \*root)

{

    Stack s1;

    Stack s2;

    BSTNode \*temp;

    s1.top = NULL;

    s2.top = NULL;

    temp = root;

    if(temp == NULL){}

    else{

        push(&s1, temp);

        while(!isEmpty(&s1)){

            temp = pop(&s1);

            push(&s2,temp);

            if(temp->left != NULL) push(&s1, temp->left);

            if(temp->right != NULL) push(&s1, temp->right);

        }

        while(!isEmpty(&s2)){

            temp = pop(&s2);

            printf("%d",temp->item);

        }

    }

}