DSAL

Week 8

Udeet Mittal

CSE C3

Roll Number 64

I. SOLVED EXERCISE:

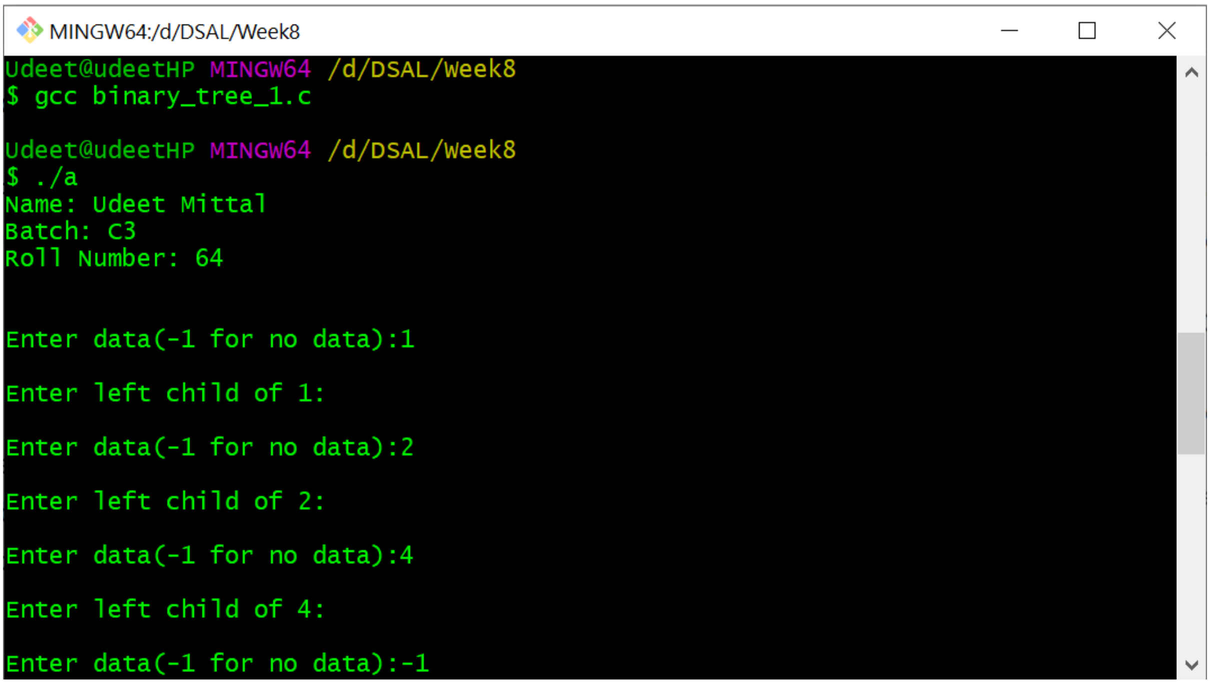
1) Create a binary tree using recursion and display its elements using all the traversal methods.

File name: binary_tree_1.c

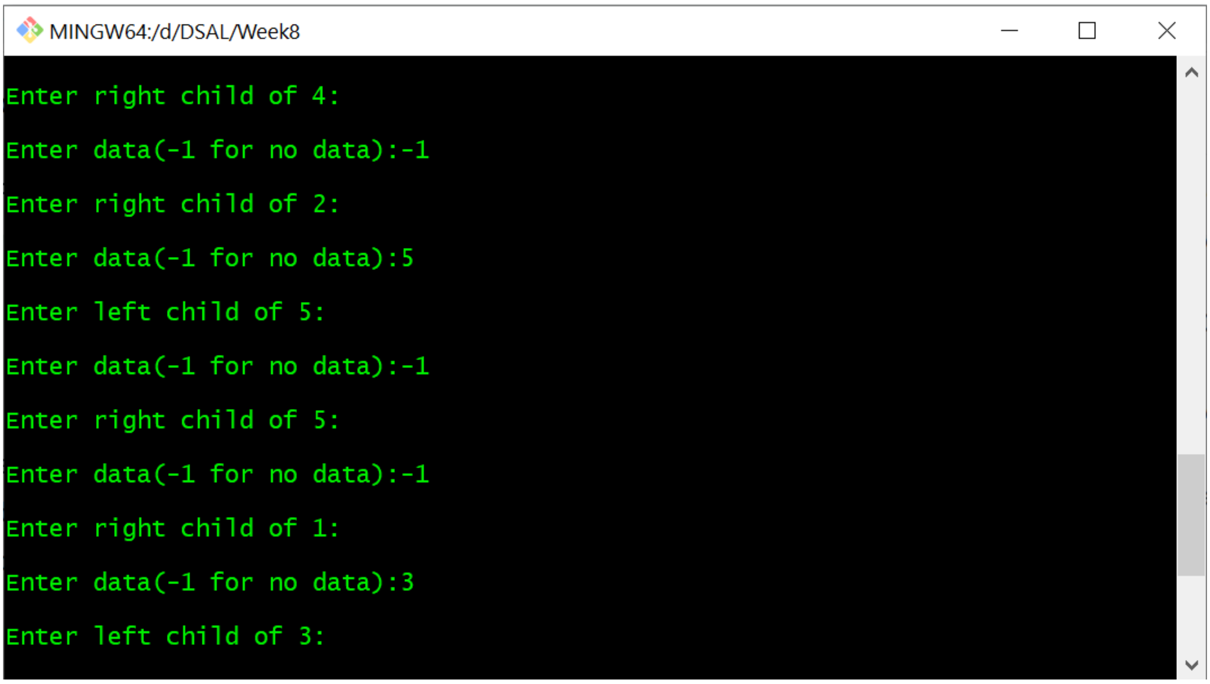
```
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
   int data;
   struct node *left;
   struct node *right;
} node;

node *create()
{
   node *p;
   int x;
   printf("\nEnter data(-1 for no data):");
   scanf("%d", &x);
   if (x == -1)
```

```
        return NULL;
    p = (node *)malloc(sizeof(node));
    p->data = x;
    printf("\nEnter left child of %d:\n", x);
    p->left = create();

    printf("\nEnter right child of %d:\n", x);
    p->right = create();
    return p;
}

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

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

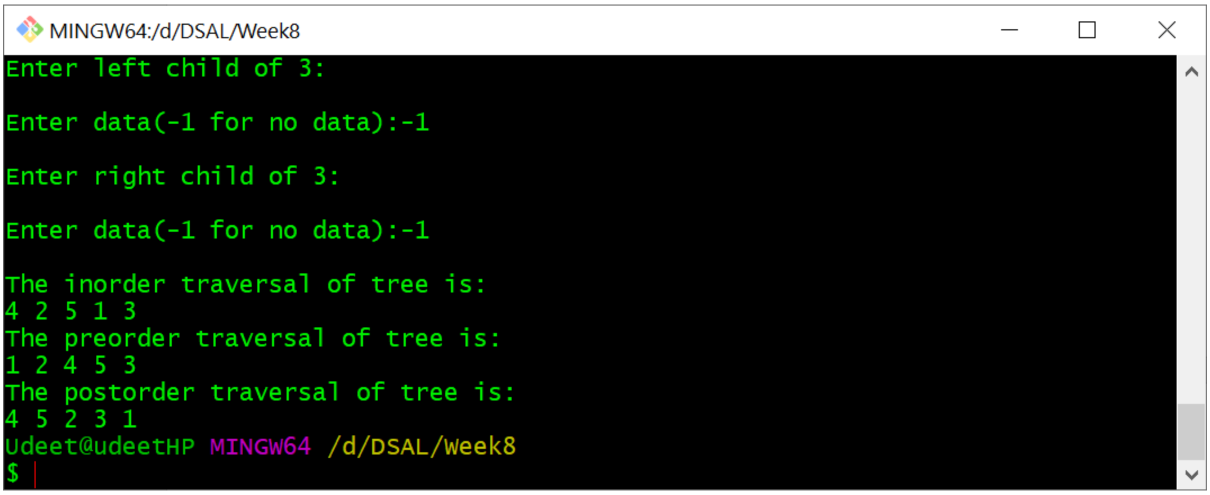
```
}

void preorder(node *t)
{
    if (t != NULL)
    {
        printf("%d ", t->data);
        preorder(t->left);
        preorder(t->right);
    }
}

int main()
{
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n\n");
    node *root;
    root = create();
    printf("\nThe inorder traversal of tree is:\n");
    inorder(root);
    printf("\nThe preorder traversal of tree is:\n");
    preorder(root);
    printf("\nThe postorder traversal of tree is:\n");
    postorder(root);
    return 0;
}
```



```
Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ gcc binary_tree_1.c

Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64


Enter data(-1 for no data):1

Enter left child of 1:

Enter data(-1 for no data):2

Enter left child of 2:

Enter data(-1 for no data):4

Enter left child of 4:

Enter data(-1 for no data):-1
```



```
Enter right child of 4:

Enter data(-1 for no data):-1

Enter right child of 2:

Enter data(-1 for no data):5

Enter left child of 5:

Enter data(-1 for no data):-1

Enter right child of 5:

Enter data(-1 for no data):-1

Enter right child of 1:

Enter data(-1 for no data):3

Enter left child of 3:
```



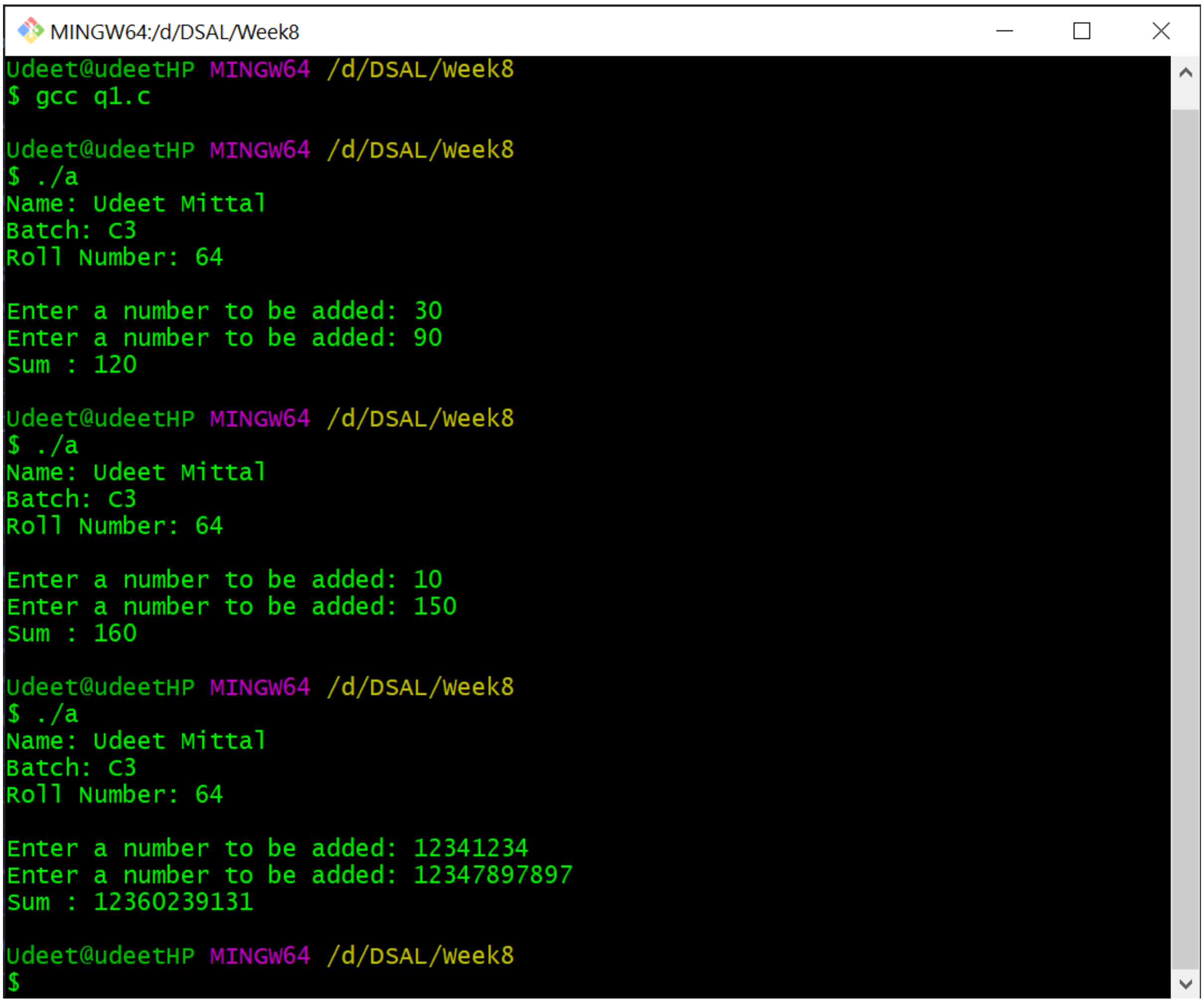
```
Enter left child of 3:

Enter data(-1 for no data):-1

Enter right child of 3:

Enter data(-1 for no data):-1

The inorder traversal of tree is:
4 2 5 1 3
The preorder traversal of tree is:
1 2 4 5 3
The postorder traversal of tree is:
4 5 2 3 1
Udeet@udeetHP MINGW64 /d/DSAL/Week8
$
```

Questions for Lab8

1) Add two long positive integers represented using circular doubly linked list with header node.

Filename: q1.c

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct node *nodeptr;

typedef struct node

{

   nodeptr rlink, llink;

   long data;

} node;

nodeptr create()

{

   nodeptr temp = malloc(sizeof(node));

   return temp;

}

void insert(nodeptr *n, long x)

{
   if (*n == NULL)
   {
       *n = create();
       (*n)->data = x;
       (*n)->llink = (*n)->rlink = *n;
   }
   else
   {
       nodeptr temp = *n;
       while (temp->llink != *n)
           temp = temp->llink;

       nodeptr newnode = create();
       newnode->data = x;
       temp->llink = newnode;
       newnode->rlink = temp;
       newnode->llink = *n;
       (*n)->rlink = newnode;
   }
}
nodeptr readlong()
{
   nodeptr head;
   char str[100];
   int i;
   printf("Enter a number to be added: ");
   scanf("%s", str);
   nodeptr n = create();
   n->llink = n->rlink = n;
   for (i = 0; str[i]; i++)
       insert(&n, str[i] - '0');
   return n;
}
nodeptr addlong(nodeptr A, nodeptr B)
{
   long digit, sum, carry = 0;
   nodeptr head, r, R, a, b;
   a = A->rlink;
   b = B->rlink;
   head = create();
   head->llink = head->rlink = head;
   while (a != A && b != B)
   {
       sum = a->data + b->data + carry;
       digit = sum % 10;
       carry = sum / 10;
       insert(&head, digit);
       a = a->rlink;
       b = b->rlink;
   }
   if (a != A)
   {
       r = a;
       R = A;
   }
   else
   {
       r = b;
       R = B;
   }
   while (r != R)
   {
       sum = r->data + carry;
       digit = sum % 10;
       carry = sum / 10;
       insert(&head, digit);
       r = r->rlink;
   }
   if (carry)
       insert(&head, carry);
   return head;
}
void display(nodeptr *n)
{
   for (nodeptr temp = (*n)->rlink; temp != *n; temp = temp->rlink)
       printf("%ld", temp->data);
   printf("\n");
}
int main()
{
   printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n\n");
   nodeptr A, B, sum;
   A = readlong();
   B = readlong();
   sum = addlong(A, B);
   printf("Sum : ");
   display(&sum);
   return 0;

}
```



```
Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ gcc q1.c

Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter a number to be added: 30
Enter a number to be added: 90
Sum : 120

Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter a number to be added: 10
Enter a number to be added: 150
Sum : 160

Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter a number to be added: 12341234
Enter a number to be added: 12347897897
Sum : 12360239131

Udeet@udeetHP MINGW64 /d/DSAL/Week8
$
```

2) Write a menu driven program to do the following using iterative functions:

i) To create a BST for a given set of integer numbers

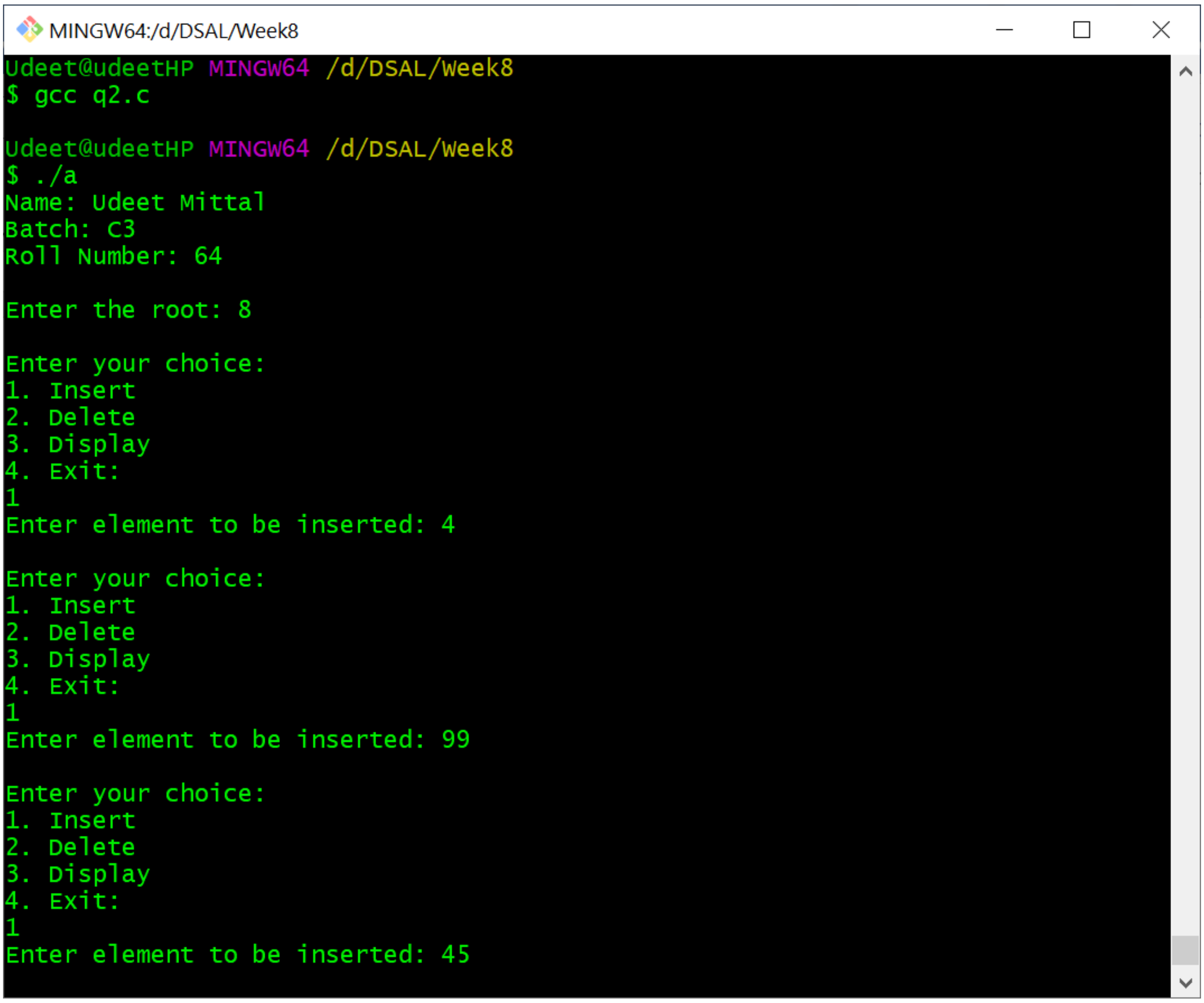
ii) To delete a given element from BST.

iii) Display the elements using iterative in-order traversal.

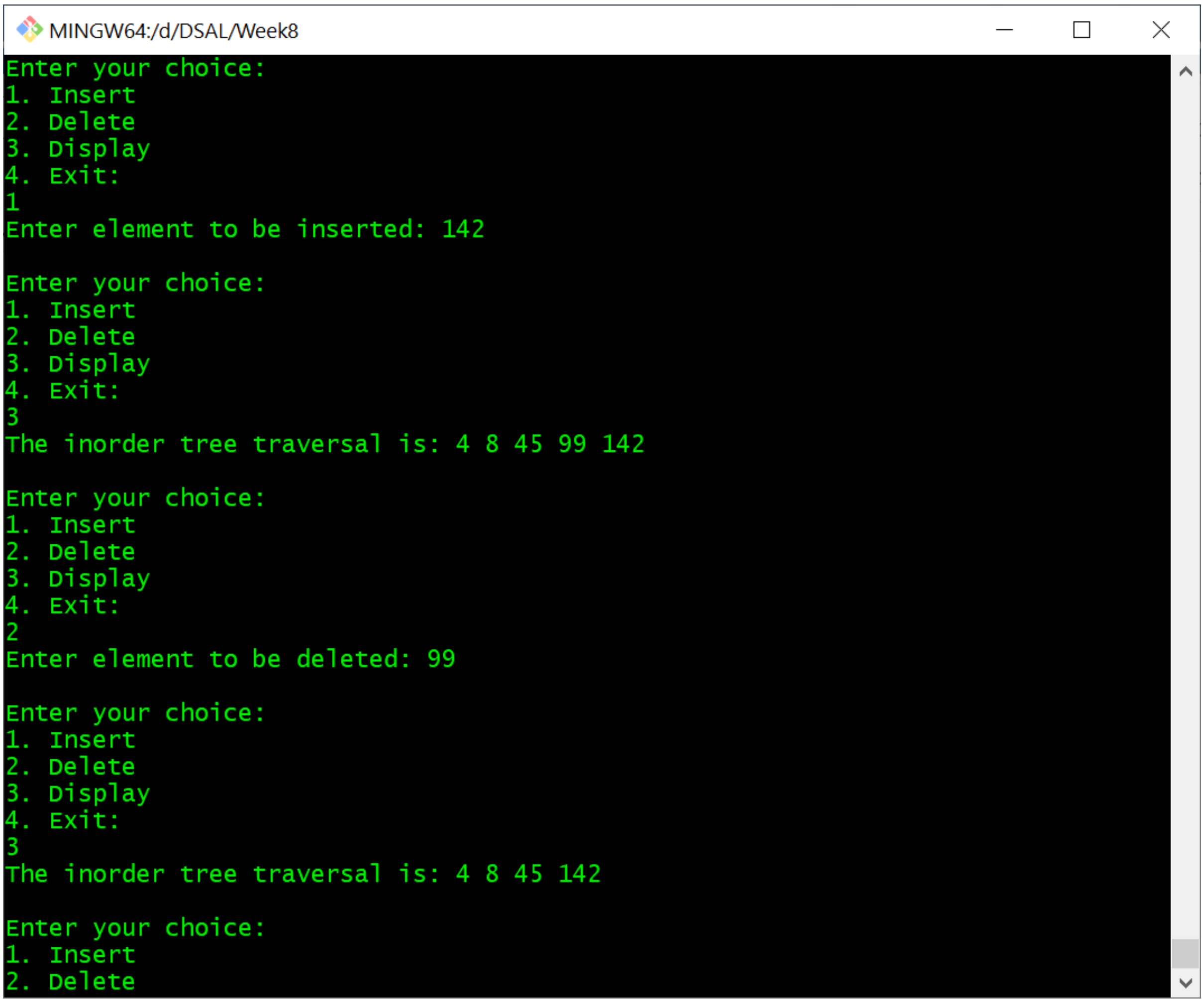
Filename: q2.c

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 10
```

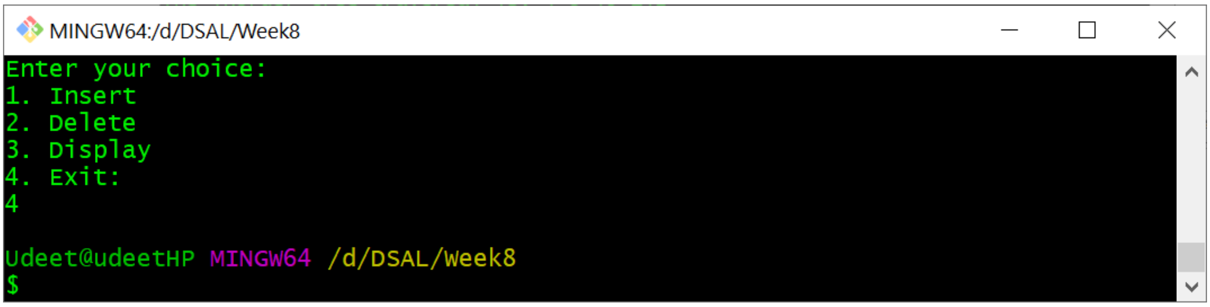
```
typedef struct node
{
    int key;
    struct node *left, *right;
} * NODE;
typedef struct
{
    NODE S[MAX];
    int tos;
} STACK;
NODE newNODE(int item)
{
    NODE temp = (NODE)malloc(sizeof(struct node));
    temp->key = item;
    temp->left = temp->right = NULL;
    return temp;
}
void push(STACK *s, NODE n)
{
    s->S[++(s->tos)] = n;
}
NODE pop(STACK *s)
{
    return s->S[(s->tos)--];
}
void inorder(NODE root)
{
    NODE curr;
    curr = root;
    STACK S;
    S.tos = -1;
    push(&S, root);
    curr = curr->left;
    while (S.tos != -1 || curr != NULL)
    {
        while (curr != NULL)
        {
            push(&S, curr);
            curr = curr->left;
        }
        curr = pop(&S);
        printf("%d ", curr->key);
        curr = curr->right;
    }
}
NODE insert(NODE node, int key)
{
    if (node == NULL)
        return newNODE(key);
    if (key < node->key)
        node->left = insert(node->left, key);
    else if (key > node->key)
        node->right = insert(node->right, key);
    return node;
}
NODE minValueNode(NODE node)
{
    NODE current = node;
    while (current && current->left != NULL)
        current = current->left;
    return current;
}
NODE deleteNode(NODE root, int key)
{
    if (root == NULL)
        return root;
    if (key < root->key)
        root->left = deleteNode(root->left, key);
    else if (key > root->key)
        root->right = deleteNode(root->right, key);
    else
    {
        if (root->left == NULL)
        {
            NODE temp = root->right;
            free(root);
            return temp;
        }
        else if (root->right == NULL)
        {
            NODE temp = root->left;
            free(root);
            return temp;
        }
        NODE temp = minValueNode(root->right);
        root->key = temp->key;
        root->right = deleteNode(root->right, temp->key);
    }
    return root;
}
int main()
{
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n\n");
    NODE root = NULL;
    int k;
    printf("Enter the root: ");
    scanf("%d", &k);
    root = insert(root, k);
    int ch;
    do
    {
        printf("\nEnter your choice:");
        printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit:\n");
        scanf("%d", &ch);
        switch (ch)
        {
        case 1:
            printf("Enter element to be inserted: ");
            scanf("%d", &k);
            root = insert(root, k);
            break;

        case 2:
            printf("Enter element to be deleted: ");
            scanf("%d", &k);
            root = deleteNode(root, k);
            break;
        case 3:printf("The inorder tree traversal is: ");
            inorder(root);
            printf("\n");
            break;
        }
    } while (ch < 4);
    return 0;
}
```



```
Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ gcc q2.c

Udeet@udeetHP MINGW64 /d/DSAL/Week8
$ ./a
Name: Udeet Mittal
Batch: C3
Roll Number: 64

Enter the root: 8

Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
1
Enter element to be inserted: 4

Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
1
Enter element to be inserted: 99

Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
1
Enter element to be inserted: 45
```



```
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
1
Enter element to be inserted: 142

Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
3
The inorder tree traversal is: 4 8 45 99 142

Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
2
Enter element to be deleted: 99

Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
3
The inorder tree traversal is: 4 8 45 142

Enter your choice:
1. Insert
2. Delete
```



```
Enter your choice:
1. Insert
2. Delete
3. Display
4. Exit:
4

Udeet@udeetHP MINGW64 /d/DSAL/Week8
$
```