```
/*
Subject: Data Structures Laboratory
Pr No: 10
Title: Read the marks obtained by students of second year in an online examination
       of particular subject. Find out maximum and minimum marks obtained in that subject.
       Use heap data structure. Analyze the algorithm.
       Inputs: Marks of 07 Students.
       Output: a) Create BST and display.
               b) Convert BST into MAX Heap Tree and display.
               c) Display Maximum Marks.
*/

//-------------------------------------------------Header Files

#include<iostream>
using namespace std;


//----------------------------------Node structure for BST
struct Node
{
   int marks;
   struct Node *left;
   struct Node *right;

}*root;

//------------------------------------------Create Node Function


struct Node* createNode()
{
   struct Node *tmp;

   tmp = new struct Node;

   cout<<"\n\t Enter Marks: ";
   cin>>tmp->marks;
   tmp->left = NULL;
   tmp->right = NULL;

   return tmp;
}

//-------------------------------------------------Create Root Function

void create_BST()
{
   if(root == NULL)
   {
         root = createNode();
         cout<<"\n\t ** Root of BST Tree is created ** ";
```

```
    }
}

//-------------------------------------------------Insert Nodes in BST Function

void insert(struct Node *Root, struct Node *newnode)
{
    if(newnode->marks < Root->marks)
    {
            if(Root->left == NULL)
              Root->left = newnode;
            else
              insert(Root->left , newnode);
    }
    else
    {
            if(Root->right == NULL)
              Root->right = newnode;
            else
              insert(Root->right , newnode);
    }
}

//-------------------------------------------------Preorder Traversal Function

void preorder(struct Node *Root)
{
    if(Root != NULL)
    {
            cout<<"  "<<Root->marks;
            preorder(Root->left);
            preorder(Root->right);
    }

}

                //.....Function to Convert BST into MAX Heap Tree
void MaxHeapify(struct Node *Parent)
{
    int val;

               val = Parent->right->marks;
       Parent->right->marks = Parent->marks;
              Parent->marks = val;
}


//.....................................................Main Function
int main()
{
    int cnt , i, j;
    struct Node *newnode;
    cout<<"\n\n --------***Create and Display MAX Heap Tree***-------";

    root = NULL;
    i = j = 0;


    create_BST();                   //.....Step 1: Create BST
                  //.....Step 2: Insert Nodes in BST
    cout<<"\n\n How many nodes to insert? : ";
    cin>>cnt;

    for(i=0; i<cnt; i++)
    {
         newnode = createNode();
         insert(root , newnode);
    }
                //.....Step 3: Display BST
    cout<<"\n\n Preorder Traversal: ";
    preorder(root);

                //.....Step 4: Convert BST into MAX Heap Tree
    MaxHeapify(root);
    MaxHeapify(root->left);
    MaxHeapify(root->right);
    MaxHeapify(root);
              //.....Step 5: Display MAX Heap Tree
    cout<<"\n\n After Heapifying BST.........";
    cout<<"\n\n Max Heap Tree Preorder Traversal: ";
    preorder(root);

    cout<<"\n\n\t Maximum Marks: "<<root->marks;

    cout<<"\n\n";

    return 0;
}

/*-----------------OUTPUT---------------------
```

```
 --------***Create and Display MAX Heap Tree***-------
        Enter Marks: 30

        ** Root of BST Tree is created **

 How many nodes to insert? : 6

        Enter Marks: 20

        Enter Marks: 25

        Enter Marks: 10
        Enter Marks: 40

        Enter Marks: 35

        Enter Marks: 55

 Preorder Traversal:   30  20  10  25  40  35  55

 After Heapifying BST.........

 Max Heap Tree Preorder Traversal:   55  25  10  20  40  35  30

        Maximum Marks: 55


...Program finished with exit code 0
Press ENTER to exit console.
...............................................*/
```