"DATA - STRUCTURES"

"HOME - ASSIGNMENT - 3"

CODE : 20 IT 3303

Submitted By

M. J.N.V. Sai

208W1A12A0

IT- B

Submitted To:

V. Radheshyam

Assistant - Professor

IT- DEPARTMENT.

M.T.N.Y.Sai
208W1A12A0
IT-B

"DATA-STRUCTURES"

"HOME-ASSIGNMENT-3"

check If a Binary Tree is Symmetric or not Symmetric,

Symmetric Binary Tree:

→ If The Tree has a Symmetric Structure If The Left and Right Sub-Trees mirror at Each-other.

→ 2 Trees Mirror Each other If all The following conditions are Satisfied.

1) Both Trees are Empty, or both are non-Empty.
2) The Left Sub-Tree is The Mirror of The Right Sub-Tree.
3) The Right Sub-Tree is The mirror of Left Sub-Tree.

program:

```
#include < ~~Stdio.h~~ iostream >
~~#include <Stdlib.h>~~
Using namespace std;
Struct Tree
{
    char data;
    Tree *left;
    Tree *right;
    Tree (char data)
    {
        this->data = data;
        left = NULL;
        right = NULL;
    }
};
```

```
bool  symmetricBT (Tree  *root_s1 , Tree  * root_s2)
{
    If ( !root_s1  &&  !root_s2)
     {
         return true;
      }
     else
     {
       If (root_s1 && root_s2)
        {
           If (root_s1 -> data  = root_s2 -> data)
           {
               return SymmetricBT (root_s1 -> left, root_s2 -> rig
                   SymmetricBT (root_s1 -> right, root_s2 -> lef
           }
        }
       return false;
     }
}
int main ()
{
    Tree  *root  =  new Tree ('1');
    root -> left  =  new  Tree  ('3');
    root -> right  =  new  Tree  ('3');
    root -> left -> left = new Tree ('4');
    root -> left -> Right = new Tree ('6');
    root -> Right -> left = new Tree ('6');
    root -> right -> right = new Tree ('4');
    If (SymmetricBT (root, root))
    {
        Cout << "The Binary Tree is Symmetric. " << Endl;
    }
    Else
    {
        Cout << " the Binary Tree is asymmetric. " << Endl;
    }
    return 0;
}
```

OUTPUT:

```
The Binary Tree is Symmetric.
```