~\Desktop\Outputs\HPC\1-A.c

```
Name:Arshin Sajid Mokashi
Roll NO.:COBB26

#CODE
#include<iostream>
#include<stdlib.h>
#include<queue>
using namespace std;

class node
{
   public:
    node *left, *right;
    int data;
};

class Breadthfs
{
 public:
 node *insert(node *, int);
 void bfs(node *);
};

node *insert(node *root, int data)
{
    if(!root)
    {
     root=new node;
     root->left=NULL;
     root->right=NULL;
     root->data=data;
     return root;
    }

    queue<node *> q;
    q.push(root);

    while(!q.empty())
    {
     node *temp=q.front();
     q.pop();

     if(temp->left==NULL)
     {
         temp->left=new node;
         temp->left->left=NULL;
         temp->left->right=NULL;
         temp->left->data=data;
         return root;
     }
     else
```

```
50      {
51      q.push(temp->left);
52      }
53
54      if(temp->right==NULL)
55      {
56          temp->right=new node;
57          temp->right->left=NULL;
58          temp->right->right=NULL;
59          temp->right->data=data;
60          return root;
61      }
62      else
63      {
64      q.push(temp->right);
65      }
66     }
67 }
68
69 void bfs(node *head)
70 {
71      queue<node*> q;
72      q.push(head);
73      int qSize;
74      while (!q.empty())
75      {
76          qSize = q.size();
77          #pragma omp parallel for
78          for (int i = 0; i < qSize; i++)
79          {
80              node* currNode;
81              #pragma omp critical
82              {
83                currNode = q.front();
84                q.pop();
85                cout<<"\t"<<currNode->data;
86              }
87              #pragma omp critical
88              {
89              if(currNode->left)
90                  q.push(currNode->left);
91              if(currNode->right)
92                  q.push(currNode->right);
93              }
94          }
95      }
96 }
97
98 int main(){
99     node *root=NULL;
100    int data;
101    char ans;
102
```

```
    do
    {
     cout<<"\n enter data=>";
     cin>>data;
     root=insert(root,data);
     cout<<"do you want insert one more node?";
     cin>>ans;
    }while(ans=='y'||ans=='Y');

    bfs(root);
    return 0;
}

/*

OUTPUT

enter data=>10
do you want insert one more node? y
 enter data=>20
do you want insert one more node? y
 enter data=>30
do you want insert one more node? y
 enter data=>40
do you want insert one more node? n
    10  20  30  40

    [10]
    /    \
 [20]    [30]
 /
[40]

*/
```