PRACTICAL NO: 1A

TITLE: Design and implement Parallel Breadth First Search based on existing algorithms using OpenMP. Use a Tree or an undirected graph for BFS

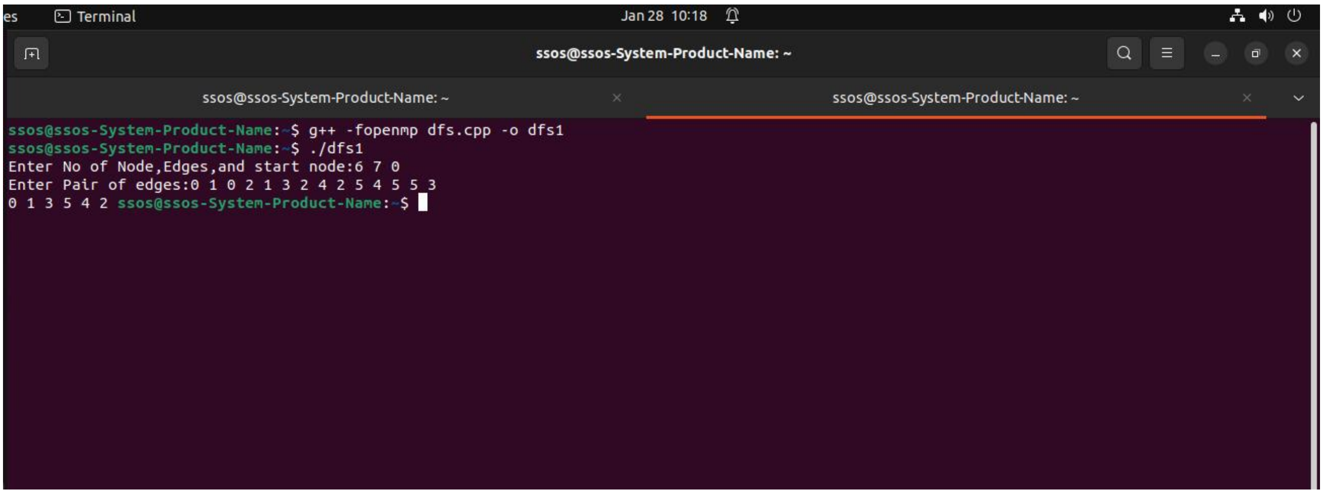
CODE:

```
#include <cstddef>
#include <iostream>
#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
{
        q.push(temp -> left);
}
if(temp -> right == NULL)
{
        temp -> right = new node;
        temp -> right -> left = NULL;
        temp -> right -> right = NULL;
        temp -> right -> data = data;
        return root;
}
else
```

```
{
        q.push(temp -> right);
}
}
}
void bfs(node *head)
{
        queue<node *> q;
        q.push(head);
        int qSize;
        while(!q.empty())
        {
                qSize = q.size();
                for (int i=0; i<qSize; i++)
                {
                        node* currNode;
                        {
                                currNode = q.front();
                                q.pop();
                                cout<<"\t"<<currNode -> data;
                        }
                        {
                        if(currNode -> left)
                                q.push(currNode -> left);
                        if (currNode -> right)
                                q.push(currNode -> right);
                }
                }
        }
```

```
}
int main()
{
        node *root = NULL;
        int data;
        char ans;
        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:


es
Terminal
Jan 28 10:18
ssos@ssos-System-Product-Name: ~
ssos@ssos-System-Product-Name: ~
ssos@ssos-System-Product-Name: ~

```
ssos@ssos-System-Product-Name:~$ g++ -fopenmp dfs.cpp -o dfs1
ssos@ssos-System-Product-Name:~$ ./dfs1
Enter No of Node,Edges,and start node:6 7 0
Enter Pair of edges:0 1 0 2 1 3 2 4 2 5 4 5 5 3
0 1 3 5 4 2 ssos@ssos-System-Product-Name:~$
```