**Lab-1**

**ARTIFICIAL INTELLIGENCE**

****

**Title:** - **Implement Uninformed searching Technique/s**

**(BFS - Breadth First Search)**

**Group Details:**

**Group No: TY-12 Div.: B Batch: B-3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Name** | **Roll.No** | **Gr.No** |
| 1 | Shubham Kakde | 49 | 11811244 |
| 2 | Sidharth Kakde | 50 | 11811170 |
| 3 | Vedant Katti | 55 | 11810142 |
| 4 | Vinayak Kokane | 58 | 11810499 |
| 5 | Dhanraj Kore | 60 | 11810970 |

**Implementation:**

#include <bits/stdc++.h>

using namespace std;

struct bstnode{

int data;

bstnode\* left;

bstnode\* right;

};

bstnode\* Newnode(int data){

    bstnode\* node = new bstnode();

    node->data=data;

    node->left=NULL;

    node->right=NULL;

    return node;

}

bstnode\* Insert(bstnode\* root, int data){

    if(root==NULL)

            root = Newnode(data);

    else if(data <= root->data)

            root->left = Insert(root->left,data);

    else

            root->right = Insert(root->right,data);

    return root;

}

void bfs(bstnode\* root){

     queue <bstnode\*> q;

    if(root==NULL)

        return;

    else{

    q.push(root);

    while(!q.empty()){

        bstnode\* curr=q.front();

        if(curr->left!=NULL)

            q.push(curr->left);

         if(curr->right!=NULL)

            q.push(curr->right);

        cout<<curr->data<<" ";

        q.pop();

        }

    }

}

int main(){

     bstnode\* node = NULL;

     node = Insert(node,15);

     node =Insert(node,25);

     node =Insert(node,13);

     node =Insert(node,12);

     node =Insert(node,11);

     node =Insert(node,18);

     node =Insert(node,17);

     node =Insert(node,14);

     node =Insert(node,26);

     node =Insert(node,16);

     node =Insert(node,28);

        /\*       15

               /    \

              13     25

             /  \    / \

            12  14  18  26

           /       /     \

          11      17     28

           \*/

        cout<<" The BFS traversal is : ";

        bfs(node);

}

**Graph:**

        15

               /     \

              13     25

             /  \    / \

            12  14  18  26

           /       /     \

          11      17     28

**O/P:**

