DATA STRUCTURES & ALOGRITHM MINIPROJECT

GARV SOLANKI A048 45207210044
CHESTA JANI A018

What is a general tree ?

A general tree data structure has no restriction on the number of nodes. It means that a parent node can have any number of child nodes.

In General Tree (Each node can have arbitrary number of children) Level Order Traversal

To print all of the nodes from the below general tree

```
           10
     /   /    \   \
   2  34    56   100
  / \        |   / | \
77  88      1   7  8  9
```

We Start with pushing root node in a queue and for each node we pop it, print it and push all its child in the queue.

In case of a generic tree we store child nodes in a vector. Thus we put all elements of the vector in the queue.

Below is the program to create the above tree :

Program to create and display general tree .

Code :

```
#include <bits/stdc++.h>

using namespace std;
```

```
struct Node
{
    int key;
    vector<Node *>child;
};
Node *newNode(int key)
{
    Node *temp = new Node;
    temp->key = key;
    return temp;
}

// Prints the n-ary tree level wise
void LevelOrderTraversal(Node * root)
{
    if (root==NULL)
        return;

    // Standard level order traversal code
    // using queue
    queue<Node *> q;  // Create a queue
    q.push(root); // Enqueue root
    while (!q.empty())
    {
        int n = q.size();

        // If this node has children
        while (n > 0)
        {
            // Dequeue an item from queue and print it
            Node * p = q.front();
            q.pop();
            cout << p->key << " ";

            // Enqueue all children of the dequeued item
            for (int i=0; i<p->child.size(); i++)
                q.push(p->child[i]);
            n--;
        }

        cout << endl; // Print new line between two levels
    }
}
int main()
{
    /*   creating below tree
```

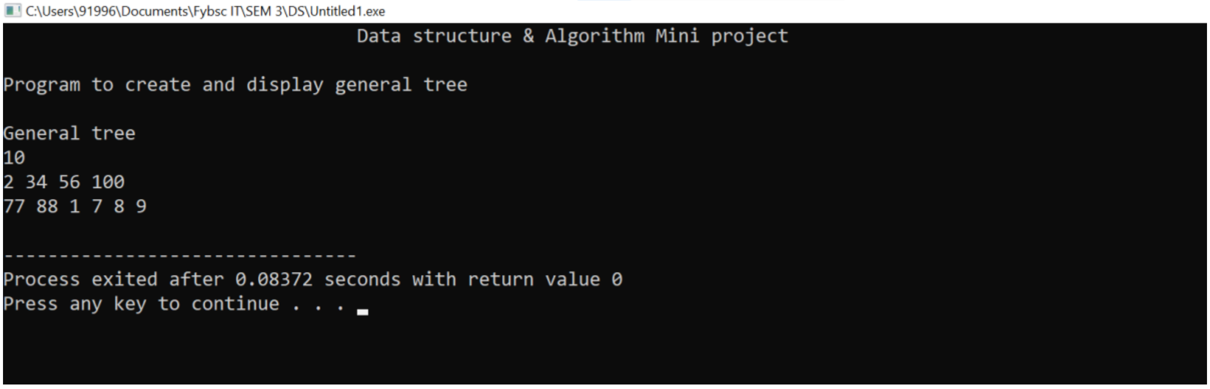
```
*              10
*        /   /    \   \
*         2  34    56   100
*       / \         |   /  | \
*      77  88       1   7  8  9
*/
Node *root = newNode(10);
(root->child).push_back(newNode(2));
(root->child).push_back(newNode(34));
(root->child).push_back(newNode(56));
(root->child).push_back(newNode(100));
(root->child[0]->child).push_back(newNode(77));
(root->child[0]->child).push_back(newNode(88));
(root->child[2]->child).push_back(newNode(1));
(root->child[3]->child).push_back(newNode(7));
(root->child[3]->child).push_back(newNode(8));
(root->child[3]->child).push_back(newNode(9));



  cout << "Data structure & Algorithm Mini project \n";
  cout << "\nProgram to create and display general tree \n";
cout << "\nGeneral tree \n";
LevelOrderTraversal(root);
```

```
    return 0;
}
```

Output :



C:\Users\91996\Documents\Fybsc IT\SEM 3\DS\Untitled1.exe

```
                                Data structure & Algorithm Mini project

Program to create and display general tree

General tree
10
2 34 56 100
77 88 1 7 8 9

--------------------------------
Process exited after 0.08372 seconds with return value 0
Press any key to continue . . .
```