

Group B1

Title:- Use Adjacency List And Matrix Representation of graph.

```
#include<iostream>
#include<stdlib.h>
using namespace std;
#define SIZE 3
struct GraphNode
{
    int Vertex;
    GraphNode *next;
};
class Graph
{
    int n=SIZE;
public:
    GraphNode *Head[SIZE]={NULL};
    int Array[SIZE][SIZE];
    void Create();
    void ListIndegree(int);
    void ListOutdegree(int);
    void List_in(int);
    void MatrixInsert();
    void MatrixIndegree(int);
    void MatrixOutdegree(int);
    void Show();
    void MatrixShow();
};
void Graph::List_in(int vertex)
{
    int i=0;
    int deg=0;
    GraphNode *temp;
    temp=Head[vertex];
    for(i=0;i<SIZE;i++){
        while (temp->next!=NULL){
            ++deg;
            temp=temp->next;
        }
    }
    cout<<deg;
}
void Graph::MatrixInsert()
{
    int n=0;
    cout<<"\nHow many Vertices";
```

```

        cin>>n;
        for(int i=1;i<=n;i++){
            for(int j=1;j<=n;j++){
                cout<<"\nEnter The Value For "<<i<<" "<<j;
                cin>>Array[i][j];
            }
        }
    }
    void Graph::MatrixIndegree(int vertex)
    {
        int Count_indegree=0;
        for(int i=0;i<SIZE;i++){
            if(Array[i][vertex]==1){
                Count_indegree+=1;
            }
        }
        cout<<"\n"<<Count_indegree;
    }
    void Graph::MatrixOutdegree(int vertex)
    {
        int Count_outdegree=0;
        for(int i=0;i<SIZE;i++){
            if(vertex==Array[vertex][i]){
                Count_outdegree+=1;
            }
        }
        cout<<"\n"<<Count_outdegree;
    }
    void Graph::ListOutdegree(int Vertex)
    {
        GraphNode *temp;
        int Count_of_out=0;
        temp=Head[Vertex];
        do{
            temp=temp->next;
            Count_of_out++;
        }while(temp->next!=NULL);
        cout<<"\n"<<Count_of_out;
    }
    void Graph::Show()
    {
        int i=1;
        GraphNode *temp;
        temp=Head[i];

```

```

for(i=1;i<=SIZE;i++){
    cout<<Head[i]->Vertex<<"->";
    temp=Head[i]->next;
    do{
        if(temp==NULL){
            cout<<"NULL";
        }
        else{
            cout<<" "<<temp->Vertex<<"->";
            temp=temp->next;
        }
    }while(temp);
    cout<<"\n";
}
}
void Graph::Create()
{
    int i=1;
    int Data;
    char choice;
    GraphNode *temp,*p;
    cout<<"\nHow Many Nodes";
    cin>>n;
    while (n){
        Head[i]=new GraphNode;
        cout<<"\nEnter the Vertex name";
        cin>>Head[i]->Vertex;
        Head[i]->next=NULL;
        temp=Head[i];
        do{
            cout<<"\nEnter the Vertex Data";
            cin>>Data;
            if(Data==1){
                break;
            }
            else{
                p = new GraphNode;
                p->Vertex=Data;
                p->next=NULL;
                temp->next=p;
                temp=p;
            }
            cout<<"\nIs There Any Adj.Vertex to "<<Head[i]->Vertex<<" (y/n)";
            cin>>choice;
        }while((choice=='y') || (choice=='Y'));
    }
}

```

```

        n--;
        i++;
    }

}

void Graph::MatrixShow()
{
    for(int i=1;i<=SIZE;i++){
        cout<<"\n"<<i<<"|";
        for(int j=1;j<=SIZE;j++){
            cout<<"\t"<<Array[i][j];
        }
    }
}

int main()
{
    Graph Graph_obj;
    int choice;
    int Vertex_Data;
    do{
        cout<<"\n\t1.Create Graph\n\t2.Find Indegree\n\t3.Find Outdegree\n\t4.Show
\n\t5.Matrix Creation\n\t6.Matrix SHow\n\t7.Matrix Indegree\n\t8.Maxtrix outDegre
e";
        cin>>choice;
        switch(choice)
        {
            case 1: Graph_obj.Create();
                    break;

            case 2: cout<<"\nEnter the Vertex Name";
                    cin>>Vertex_Data;
                    Graph_obj.List_in(Vertex_Data);
                    break;

            case 3: cout<<"\nEnter the Vertex Name";
                    cin>>Vertex_Data;
                    Graph_obj.ListOutdegree(Vertex_Data);
                    break;

            case 4: Graph_obj.Show();
                    break;

            case 5: Graph_obj.MatrixInsert();
                    break;

```

```

        case 6: Graph_obj.MatrixShow();
                break;

        case 7: cin>>Vertex_Data;
                Graph_obj.MatrixIndegree(Vertex_Data);
                break;

        case 8: cin>>Vertex_Data;
                Graph_obj.MatrixOutdegree(Vertex_Data);
                break;
    }
    }while(choice<=8);
    return 0;
}

```

Output:

1.Create Graph

2.Find Indegre

3.Find Outdegree

4.Show

5.Matrix Creation

6.Matrix SHow

7.Matrix Indegree

8.Maxtrix outDegree1

How Many Nodes4

Enter the Vertex name1

Enter the Vertex Data2

Is There Any Adj.Vertex to 1 (y/n)y

Enter the Vertex Data3

Is There Any Adj.Vertex to 1 (y/n)n

Enter the Vertex name2

Enter the Vertex Data1

Is There Any Adj.Vertex to 2 (y/n)y

Enter the Vertex Data4

Is There Any Adj.Vertex to 2 (y/n)n

Enter the Vertex name3

Enter the Vertex Data2

Is There Any Adj.Vertex to 3 (y/n)y

Enter the Vertex Data4

Is There Any Adj.Vertex to 3 (y/n)n

Enter the Vertex name4

Enter the Vertex Data1

Is There Any Adj.Vertex to 4 (y/n)y

Enter the Vertex Data3

Is There Any Adj.Vertex to 4 (y/n)n

1.Create Graph

2.Find Indegre

3.Find Outdegree

4.Show

5.Matrix Creation

6.Matrix SHow

7.Matrix Indegree

8.Maxtrix outDegree2

Enter the Vertex Name1

2

1.Create Graph

2.Find Indegre

3.Find Outdegree

4.Show

5.Matrix Creation

6.Matrix SHow

7.Matrix Indegree

8.Maxtrix outDegree3

Enter the Vertex Name2

2

1.Create Graph

2.Find Indegre

3.Find Outdegree

4.Show

5.Matrix Creation

6.Matrix SHow

7.Matrix Indegree

8.Maxtrix outDegree4

1->2->3->

2->1->4->

3->2->4->

1.Create Graph

2.Find Indegre

3.Find Outdegree

4.Show

5.Matrix Creation

6.Matrix SHow

7.Matrix Indegree

8.Maxtrix outDegree5

How many Vertices4

Enter The Value For 1 11

Enter The Value For 1 21

Enter The Value For 1 30

Enter The Value For 1 41

Enter The Value For 2 11

Enter The Value For 2 20

Enter The Value For 2 31

Enter The Value For 2 41

Enter The Value For 3 10

Enter The Value For 3 21

Enter The Value For 3 31

Enter The Value For 3 41

Enter The Value For 4 11

Enter The Value For 4 21

Enter The Value For 4 31

Enter The Value For 4 40

1.Create Graph

2.Find Indegre

3.Find Outdegree

4.Show

5.Matrix Creation

6.Matrix SHow

7.Matrix Indegree

8.Maxtrix outDegree6

1| 1 1 0

2| 1 0 1

3| 0 1 1

- 1.Create Graph
- 2.Find Indegre
- 3.Find Outdegree
- 4.Show
- 5.Matrix Creation
- 6.Matrix SHow
- 7.Matrix Indegree
- 8.Maxtrix outDegree7

1

2

- 1.Create Graph
- 2.Find Indegre
- 3.Find Outdegree
- 4.Show
- 5.Matrix Creation
- 6.Matrix SHow
- 7.Matrix Indegree
- 8.Maxtrix outDegree8

2

0

- 1.Create Graph
- 2.Find Indegre
- 3.Find Outdegree

4.Show

5.Matrix Creation

6.Matrix SHow

7.Matrix Indegree

8.Maxtrix outDegree