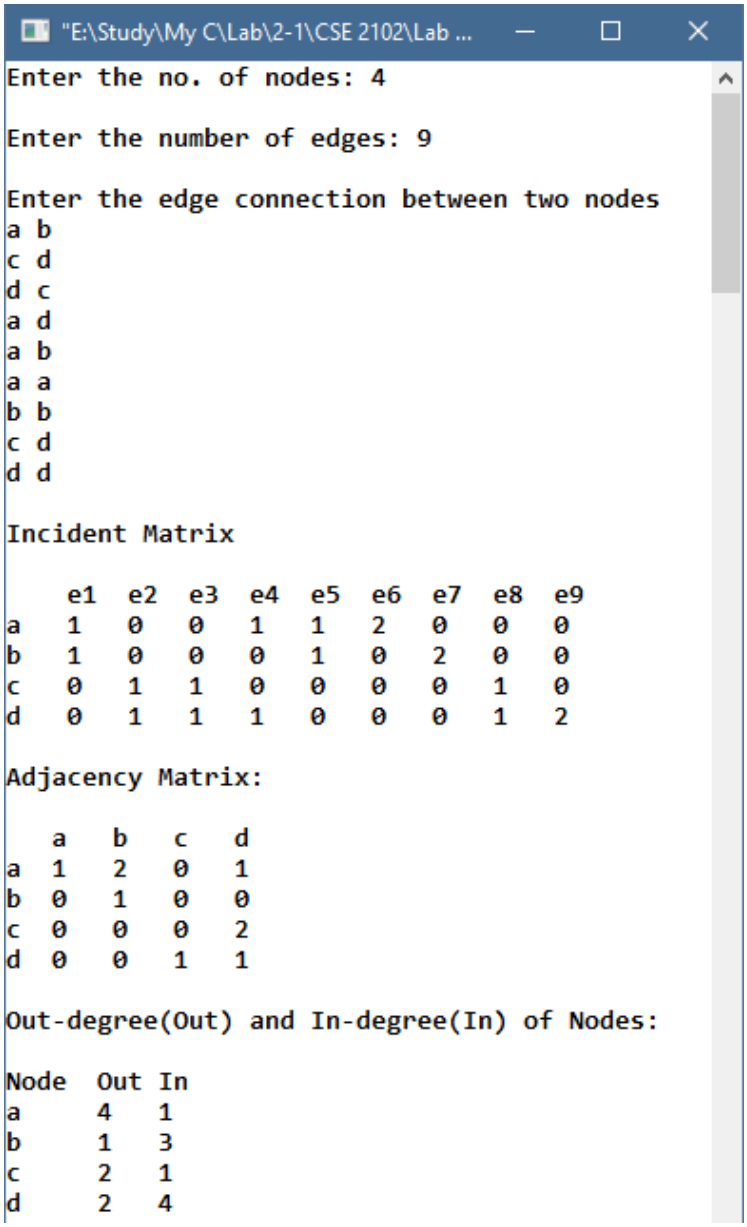
Code

```
#include<bits/stdc++.h>
using namespace std;

int n,e,A,B,E,C,D,c=0;
int X[10];
int Y[10];
char a,b;

int main()
{
   int p = 0;
   char arr[10000];

   cout<<"Enter the no. of nodes: ";
   scanf("%d",&n);

   cout<<"\nEnter the number of edges: ";
   scanf("%d,&",&e);

   int AM[n][n];
   int INC[n][e];

   for(int i=0; i<n ;i++)
      for(int j=0 ; j<n ; j++)
      AM[i][j] = 0;

   for(int i=0 ; i<n ; i++)
      for(int j=0 ; j<10 ; j++)
         INC[i][j] = 0;

   cout<<"\nEnter the edge connection between two nodes\n";

   for(int i=0 ; i<e ; i++)
   {
      scanf(" %c",&a);
      scanf(" %c",&b);

      A = a - 'a';
      B = b - 'a';
      C = a - 'a';

      AM[A][B]++;
      INC[C][p]++;
      INC[B][p]++;
      p++;
```

```
        X[A]++;
        Y[B]++;
    }
    cout<<"\nIncident Matrix\n"<<endl;
    cout<<"    ";
    for(int i=1 ; i<=e ; i++)
        cout<<"e"<<i<<"  ";
    cout<<endl;
    for(int i=0; i<n ;i++)
    {
        char ch = 'a' + i;
        cout<<ch<<"   ";
        for(int j=0 ; j<e ; j++)
            cout<<INC[i][j]<<"   ";
        cout<<endl;
     }
    cout<<"\nAdjacency Matrix: \n\n";
    cout<<"   ";
    for(int i=0; i<n ; i++)
    {
        char ch = 'a' + i;
        cout<<ch<<"   ";
    }
    cout<<endl;
    for(int i=0; i<n ;i++)
    {
        char ch = 'a' + i;
        cout<<ch<<"  ";
        for(int j=0 ; j<n ; j++)
            cout<<AM[i][j]<<"   ";
        cout<<endl;
    }
    cout<<"\nOut-degree(Out) and In-degree(In) of Nodes:\n\n";
    cout<<"Node "<<" Out"<<" In"<<endl;
    for(int i=0 ; i<n ; i++)
    {
        char ch ='a' +i;
        cout<<ch<<"     "<<X[i]<<"   "<<Y[i]<<endl;
    }
    return 0;
}
```

Output


"E:\Study\My C\Lab\2-1\CSE 2102\Lab ...
Enter the no. of nodes: 4
Enter the number of edges: 9
Enter the edge connection between two nodes
a b
c d
d c
a d
a b
a a
b b
c d
d d
Incident Matrix
e1 e2 e3 e4 e5 e6 e7 e8 e9
a 1 0 0 1 1 2 0 0 0
b 1 0 0 0 1 0 2 0 0
c 0 1 1 0 0 0 0 1 0
d 0 1 1 1 0 0 0 1 2
Adjacency Matrix:
a b c d
a 1 2 0 1
b 0 1 0 0
c 0 0 0 2
d 0 0 1 1
Out-degree(Out) and In-degree(In) of Nodes:
Node Out In
a 4 1
b 1 3
c 2 1
d 2 4