**Question1: Create a program to implement Graphs with Adjacency Matrix.**

**Code:**

static void Main(string[] args)

{

Console.WriteLine("Enter the number of nodes");

int n = int.Parse(Console.ReadLine());

string[] arr = new string[n];

for (int i = 0; i < n; i++)

{

Console.WriteLine((i+1)+" is connected to which nodes? e.g 3,5");

arr[i] = Console.ReadLine();

}

int[,] array = new int[n,n];

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

string temp = "" + (j+1);

if (arr[i].Contains(temp))

{

array[i, j] = 1;

}

else

{

array[i, j] = 0;

}

}

}

Console.Write(" ");

for (int i = 0; i < n; i++)

{

Console.Write(i+1+" ");

}

Console.WriteLine();

for (int i = 0; i < array.GetLength(0); i++)

{

Console.Write((i+1)+" ");

for (int j = 0; j < array.GetLength(1); j++)

{

Console.Write(array[i,j]+" ");

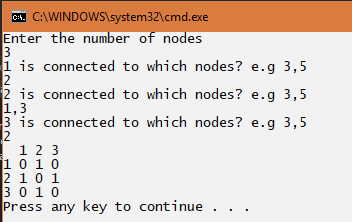
}

Console.WriteLine();

}

}

**Output:**



**Question2: Create a program to implement Graphs with Adjacency List**

**Code:**

static void Main(string[] args)

{

Console.WriteLine("Enter the number of nodes");

int n = int.Parse(Console.ReadLine());

string[] arr = new string[n];

for (int i = 0; i < n; i++)

{

Console.WriteLine((i+1)+" is connected to which nodes? e.g 3,5");

arr[i] = Console.ReadLine();

}

List<int>[] obj = new List<int>[n];

for (int i = 0; i < n; i++)

{

obj[i] = new List<int>();

string[] temp = arr[i].Split(',');

for (int a = 0; a < temp.Length; a++)

{

obj[i].Add(int.Parse(temp[a]));

}

}

for (int i = 0; i < n; i++)

{

Console.Write((i+1) +" => ");

for (int l = 0; l < obj[i].Count; l++)

{

Console.Write(obj[i][l]+" ");

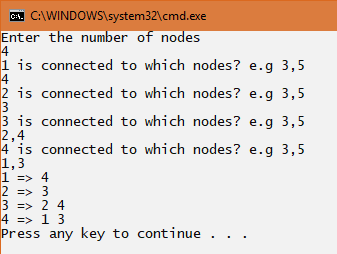
}

Console.WriteLine();

}

}

**Output:**



**Question3: Implement the Heap sort using graph concept.**

**Code:**

**Heap Class:**

class heap

{

int[] r = { 2, 5, 1, 10, 6, 9, 3, 7, 4, 8 };

public void hsort()

{

int i, t;

for (i = 5; i >= 0; i--)

{

adjust(i, 9);

}

for (i = 8; i >= 0; i--)

{

t = r[i + 1];

r[i + 1] = r[0];

r[0] = t;

adjust(0, i);

}

}

private void adjust(int i, int n)

{

int t, j;

try

{

t = r[i];

j = 2 \* i;

while (j <= n)

{

if (j < n && r[j] < r[j + 1])

j++;

if (t >= r[j])

break;

r[j / 2] = r[j];

j \*= 2;

}

r[j / 2] = t;

}

catch (IndexOutOfRangeException e)

{

Console.WriteLine("Array Out of Bounds ", e);

}

}

public void print()

{

for (int i = 0; i < 10; i++)

{

Console.WriteLine("{0}", r[i]);

}

}

**Program Class:**

static void Main(string[] args)

{

heap obj = new heap();

Console.WriteLine("Elements Before sorting : ");

obj.print();

obj.hsort();

Console.WriteLine("Elements After sorting : ");

obj.print();

Console.Read();

}

**Output:**

