```
import java.util.*;
import java.io.*;
public class two_PageRank {
public int path[][] = new int[10][10];
public double pagerank[] = new double[10];
public void calc(double totalNodes){
double InitialPageRank;
double OutgoingLinks=0;
double DampingFactor = 0.85;
double TempPageRank[] = new double[10];
int ExternalNodeNumber;
int InternalNodeNumber;
int k=1; // For Traversing
int ITERATION_STEP=1;
InitialPageRank = 1/totalNodes;
System.out.printf(" Total Number of Nodes :"+totalNodes+"\t Initia
for(k=1;k<=totalNodes;k++)
{
this.pagerank[k]=InitialPageRank;
}
System.out.printf("\n Initial PageRank Values , 0th Step \n");
for(k=1;k<=totalNodes;k++)
{
System.out.printf(" Page Rank of "+k+" is :\t"+this.pagerank[k]+"\
}
while(ITERATION_STEP<=2) // Iterations
{
for(k=1;k<=totalNodes;k++)
{
TempPageRank[k]=this.pagerank[k];
this.pagerank[k]=0;
}
for(InternalNodeNumber=1;InternalNodeNumber<=totalNodes;InternalNo
{
for(ExternalNodeNumber=1;ExternalNodeNumber<=totalNodes;ExternalNo
{
if(this.path[ExternalNodeNumber][InternalNodeNumber] == 1)
{
k=1;
OutgoingLinks=0;  // Count the Number of Outgoing Links for each E
```

```
while(k<=totalNodes)
{
if(this.path[ExternalNodeNumber][k] == 1 )
{
OutgoingLinks=OutgoingLinks+1; // Counter for Outgoing Links
}
k=k+1;
}
this.pagerank[InternalNodeNumber]+=TempPageRank[ExternalNodeNumber
}
}
}
System.out.printf("\n After "+ITERATION_STEP+"th Step \n");
for(k=1;k<=totalNodes;k++)
System.out.printf(" Page Rank of "+k+" is :\t"+this.pagerank[k]+"\
ITERATION_STEP = ITERATION_STEP+1;
}
for(k=1;k<=totalNodes;k++)
{
this.pagerank[k]=(1-DampingFactor)+ DampingFactor*this.pagerank[k]
}
System.out.printf("\n Final Page Rank : \n");
for(k=1;k<=totalNodes;k++)
{
System.out.printf(" Page Rank of "+k+" is :\t"+this.pagerank[k]+"\
}
}
public static void main(String args[])
{
int nodes,i,j,cost;
Scanner in = new Scanner(System.in);
System.out.println("Enter the Number of WebPages \n");
nodes = in.nextInt();
two_PageRank p = new two_PageRank();
System.out.println("Enter the Adjacency Matrix with 1->PATH & 0->N
for(i=1;i<=nodes;i++)
for(j=1;j<=nodes;j++)
{
p.path[i][j]=in.nextInt();
if(j==i)
p.path[i][j]=0;
}
p.calc(nodes);
}
}

// OUTPUT

// Enter the Number of WebPages

// 4
// Enter the Adjacency Matrix with 1->PATH & 0->NO PATH Between tw

// 0 1 0 1
// 1 2 3 0
// 2 1 0 1
// 0 2 1 1
//  Total Number of Nodes :4.0      Initial PageRank  of All Nodes

//  Initial PageRank Values , 0th Step
//  Page Rank of 1 is :    0.25
//  Page Rank of 2 is :    0.25
//  Page Rank of 3 is :    0.25
//  Page Rank of 4 is :    0.25

//  After 1th Step
//  Page Rank of 1 is :    0.25
//  Page Rank of 2 is :    0.25
//  Page Rank of 3 is :    0.25
//  Page Rank of 4 is :    0.25

//  After 2th Step
//  Page Rank of 1 is :    0.25
//  Page Rank of 2 is :    0.25
//  Page Rank of 3 is :    0.25
//  Page Rank of 4 is :    0.25

//  Final Page Rank :
//  Page Rank of 1 is :    0.3625000000000004
//  Page Rank of 2 is :    0.3625000000000004
```

```
    //  Page Rank of 3 is :    0.36250000000000004
    //  Page Rank of 4 is :    0.36250000000000004
```