CODE:

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

void main()

{int i,j,k,n,a,it,t,count;

float sum=0,sum1=0,s=0,s1=0;

float v,u;

//To get input

printf("\nEnter the number of nodes: ");

scanf("%d",&n);

float ip[n][n],out[n],in[n],ipt[n][n],multiply[n][n],multiply1[n][n];

printf("\nEnter the links. If there is a node from A -> B: enter 1.");

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

{a=65;

for(j=0;j<n;j++)

{printf("\n%c->%c: ",a+i,a+j);

scanf("%f",&ip[i][j]);}}

float iden[4][1]={1,1,1,1};

//To print the input

a=65;

printf("\t");

for(j=0;j<n;j++)

{printf("%c\t",a+j);}

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

{a=65;

printf("\n%c",a+i);

for(k=0;k<n;k++)

{printf("\t%.2f",ip[i][k]);}}

//to calculate ip transpose

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

{for(j=0;j<n;j++)

{ipt[i][j]=ip[j][i];}}

//To ask the number of iterations

printf("\nEnter the number of iterations desired: ");

scanf("%d",&it);

k=1;

while(k<=it)

{//To calculate indegree and outdegree

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

{out[i]=0;

in[i]=0;

for(j=0;j<n;j++)

{if(ip[i][j]==1)

{out[i]++;}

if(ip[j][i]==1)

{in[i]++;}

}}

k++;

//Calculate V, U

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

for (j = 0; j < 1; j++) {

for (t = 0; t < n; t++) {

sum = sum + ipt[i][t]\*iden[t][j];}

multiply[i][j] = sum;

sum = 0;}}

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

for (j = 0; j < 1; j++) {

for (t = 0; t < n; t++) {

sum1= sum1 + ip[i][t]\*multiply[t][j];}

multiply1[i][j] = sum1;

sum1 = 0 } }

//To calculate new authority and hub

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

{for(j=0;j<1;j++)

{s=s+(multiply[i][j]\*multiply[i][j]);

s1=s1+(multiply1[i][j]\*multiply1[i][j]);

}}

printf("\nNew Hub: ");

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

{for(j=0;j<1;j++)

{multiply[i][j]=multiply[i][j]/sqrt(s);

printf("\n%c=%.2f",65+i,multiply[i][j]);

}}

printf("\nNew Auhtority:");

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

{for(j=0;j<1;j++)

{multiply1[i][j]=multiply1[i][j]/sqrt(s1);

printf("\n%c=%.2f",65+i,multiply1[i][j]);}}}}

OUTPUT:

Enter the number of nodes: 4

Enter the links. If there is a node from A -> B: enter 1.

A->A: 0

A->B: 1

A->C: 1

A->D: 1

B->A: 0

B->B: 0

B->C: 1

B->D: 1

C->A: 1

C->B: 0

C->C: 0

C->D: 1

D->A: 0

D->B: 0

D->C: 0

D->D: 1

Input Matrix is:

A B C D

A 0.00 1.00 1.00 1.00

B 0.00 0.00 1.00 1.00

C 1.00 0.00 0.00 1.00

D 0.00 0.00 0.00 1.00

Enter the number of iterations desired: 1

New Hub:

A=0.21

B=0.21

C=0.43

D=0.85

New Authority:

A=0.62

B=0.53

C=0.45

D=0.36