

ECSE215L: DATA STRUCTURES USING C++ CODATHON - II	
Date: 30-Oct-2021 9:00 AM	No. of Questions: 1
Proposed Time: 60 Minutes	Skills Tested: Matrix Handling, Logic Building, Repeatedly Min or Max Finding, String Handling.

PRODUCT RECOMMENDER SYSTEM

Mr. XYZ is appointed as a manager of an online shopping site. At the backend, the website is maintaining the data for two main entities namely, Customers and Products. Within the few days of the manager's joining, he came to know about the challenge of decreasing sales of Products. Based on his knowledge and market survey analysis, he proposed to implement a recommender system on the website, that will utilize the Customers and Products data at the backend and will recommend Products to the Customers efficiently. After the approval of the proposed idea, the manager contacted Mr. ABC for efficient logic development. Once the logic was developed the manager approached you. As you are an efficient software engineer in the company, you are given the task of implementing the efficient Product recommendation system for the website. So, you need to implement the Product recommender system as per the following logic proposed by Mr. ABC.

INPUT: The Recommender System will take the following input as data:

1. *Customer-Product Matrix:* Binary 2-dimensional matrix containing the details of which Customer has purchased which Product. Value is 1 if the customer has purchased the corresponding product, else it is zero. Here, 0th index customer is considered as C1, 1 index customer as C2, 2 index as C3 and so on. Similarly, 0th index product is considered as P1, 1 index as P2, 2 index as P3 and so on.
2. *Product-Product Matrix:* A symmetric 2-dimensional, containing the similarity index (ranging between 0 to 1 inclusive) between the products.

STEPS: The following steps are proposed for implementation of the Product recommender system:

1. *Generate Customer-Customer Similarity:* Use the Customer-Product Matrix for generating the similarity between any two customers using the dot product. This needs to be generated for every customer, against every other customer existing within the system.
2. *Handling Customers with overall Zero Similarity:* For customers having zero similarity, calculate the most frequent products list and recommend the products having maximum frequency of buying first. In case of tie, display the product with lower index first.
3. *Handling Customers with overall Non-Zero Similarity:* For customers with overall non-zero similarity, for each customer, sum up the similarity index of non-purchased products from purchased products list, using product-product matrix. The products with highest maximum similarity index are displayed first followed by others. In case of a tie, display the products with lower index first.

OUTPUT: The Output of this will be Order-wise list of all existing customers with product recommendations.

NOTE: At the end, if no product list is generated, then write NO RECOMMENDATIONS in front of that customer.

INPUT CONSTRAINTS:

1<=Customers<=25

1<=Products<=25

INPUT FORMAT:

N -> Number of Customers

M -> Number of Products

Space separated N*M values for Customer-Product Matrix

Space separated M*M Values for Product-Product Matrix

OUTPUT FORMAT:

Customer_No->Product_No<space>Product_No<space> and so on..

Otherwise

Customer_No->NO<space>RECOMMENDATIONS

SAMPLE EXAMPLE:

Two Input Matrix:

Customer-Product Matrix				
	P1	P2	P3	P4
C1	0	1	1	0
C2	1	1	1	0
C3	0	1	0	1
C4	0	0	0	0

Product-Product Matrix				
	P1	P2	P3	P4
P1	1	0.5	0	0.2
P2	0.5	1	0.3	0.1
P3	0	0.3	1	0.6
P4	0.2	0.1	0.6	1

Step 1 (Generate Customer-Customer Similarity): Using the above data, the following customer-customer data is generated:

	C1	C2	C3	C4
C1	X	2	1	0
C2	2	X	1	0
C3	1	1	X	0
C4	0	0	0	X

Customer-Customer Similarity

	P1	P2	P3	P4
C1	0	1	1	0
C2	1	1	1	0

C1-C2 Similarity

	P1	P2	P3	P4
C1	0	1	1	0
C3	0	1	0	1

C1-C3 Similarity

Step-wise Example:

Calculating C1-C2 Similarity: Use Customer-Product Matrix: $0*1+1*1+1*1+0*0=0+1+1+0=2$

Calculating C1-C3 Similarity: Use Customer-Product Matrix: $0*0+1*1+1*0+0*1=0+1+0+0=1$

And So On...

Step 2 (Handling Customers with overall Zero Similarity): For customers like Customer C4, who have overall zero similarity, calculate the most frequent product list and display the products as recommendations as shown below:

Customer-Product Matrix				
	P1	P2	P3	P4
C1	0	1	1	0
C2	1	1	1	0
C3	0	1	0	1
C4	0	0	0	0
	1	3	2	1

Here P2 (3) is the most frequent product and is followed by P3 (2). But next, we have a tie: P1 and P4 both at 1, as P1 is at smaller index so, order of consideration is: P2, P3, P1 and P4. So the list for recommendation is: P2, P3, P1, P4.

Step 3 (Handling Customers with overall Non-Zero Similarity): Here C1, C2 and C3 are the customers with overall non-zero similarity.

So, generate recommendations for C1, C2 and C3.

For C1: C1 has purchased P2 and P3. Calculate the similarity score for rest non-purchased products using Product-Product Matrix, by summing up of the similarity index.

Customer-Product Matrix				
	P1	P2	P3	P4
C1	0	1	1	0
C2	1	1	1	0
C3	0	1	0	1
C4	0	0	0	0

Product-Product Matrix				
	P1	P2	P3	P4
P1	1	0.5	0	0.2
P2	0.5	1	0.3	0.1
P3	0	0.3	1	0.6
P4	0.2	0.1	0.6	1

Customer-Product Matrix				
	P1	P2	P3	P4
C1	0	1	1	0
C2	1	1	1	0
C3	0	1	0	1
C4	0	0	0	0

Product-Product Matrix				
	P1	P2	P3	P4
P1	1	0.5	0	0.2
P2	0.5	1	0.3	0.1
P3	0	0.3	1	0.6
P4	0.2	0.1	0.6	1

Customer-Product Matrix				
	P1	P2	P3	P4
C1	0	1	1	0
C2	1	1	1	0
C3	0	1	0	1
C4	0	0	0	0

Product-Product Matrix				
	P1	P2	P3	P4
P1	1	0.5	0	0.2
P2	0.5	1	0.3	0.1
P3	0	0.3	1	0.6
P4	0.2	0.1	0.6	1

Non-purchased Products for C1: **P1** $\rightarrow (0.5+0) = 0.5$ and **P4** $\rightarrow (0.1+0.6) = 0.7$. As P4 is having higher similarity, so recommend P4 first followed by P1 to C1.

Similarly, Non-purchased Products for C2: **P4** -> $(0.2+0.1+0.6) = 0.9$. Recommend only P4 to C2.

Similarly, Non-purchased Products for C3: **P1** -> $(0.5+0.2) = 0.7$ and **P3** -> $(0.3+0.6) = 0.9$. Recommend P3 first followed by P1 to C3.

Final Output (Order-wise Customer List):

C1->P4 P1

C2->P4

C3->P3 P1

C4->P2 P3 P1 P4

NOTE: At the end, if no product list is generated, the output will be as follows. Here, CNO represents respective customer number.

CNO->NO RECOMMENDATIONS

C++ CODE SOLUTION:

```
#include <iostream>

using namespace std;

int main()
{
    int cust_n, prod_n,i=0,j=0;
    cin>>cust_n>>prod_n;
    int cust_prod[cust_n][prod_n];
    float prod_prod[prod_n][prod_n],cust_cust[cust_n][cust_n];

    //input matrix for customer-prod
    for(i=0;i<cust_n;i++)
        for(j=0;j<prod_n;j++)
            cin>>cust_prod[i][j];

    //input matrix for prod-prod
    for(i=0;i<prod_n;i++)
        for(j=0;j<prod_n;j++)
            cin>>prod_prod[i][j];

    //Generating cust-cust similarity
    for(i=0;i<cust_n;i++)
        for(j=0;j<cust_n;j++)
        {
            if(i==j)
                cust_cust[i][j]=-1;
            else
            {
                int sim=0;
                for(int k=0;k<prod_n;k++)
                {
                    sim=sim+cust_prod[i][k]*cust_prod[j][k];
                }
                cust_cust[i][j]=sim;
                cust_cust[j][i]=sim;
            }
        }

    /*
    cout<<endl<<"Cust-Cust Similarity table:"<<endl;

    for(i=0;i<cust_n;i++)
    {
```

```

        for(j=0;j<cust_n;j++)
            cout<<cust_cust[i][j]<<" ";
        cout<<endl;
    }

    cout<<endl;
    */

// Handling Overall Zero Similarity
float sum[prod_n];
int index[prod_n];
int val;
for(i=0;i<prod_n;i++)
{
    val=0;
    for(j=0;j<cust_n;j++)
    {
        val=val+cust_prod[j][i];
    }
    sum[i]=val;
    index[i]=i+1;
}

for(i = 0; i<=prod_n; i++)
{
    for(j = i+1; j<prod_n; j++)
    {
        if(sum[j] <= sum[i])
        {
            int temp = sum[i];
            sum[i] = sum[j];
            sum[j] = temp;
            temp = index[i];
            index[i] = index[j];
            index[j] = temp;
        }
    }
}

//Common Recommendations
string s;
for(i=prod_n-1;i>=0;i--)
{
    s.append("P");
    s.append(to_string(index[i]));
    s.append(" ");
}

```

```

}

//Customer-wise recommendations
for(i=0;i<cust_n;i++)
{
    int ind_buy[prod_n],ind_not_buy[prod_n],key_buy=0,key_not_buy=0;
    int cust_sum=0;
    for(j=0;j<prod_n;j++)
    {
        cust_sum=cust_sum+cust_prod[i][j];
    }
    //If Customer is New
    if(cust_sum==0)
    {
        cout<<"C"<<i+1<<"->"<<s<<endl;
    }
    else
    {
        //List of purchased and not-purchased products
        for(j=0;j<prod_n;j++)
        {
            if(cust_prod[i][j]==1)
                ind_buy[key_buy++]=j;
            else
                ind_not_buy[key_not_buy++]=j;
        }

        /*
        cout<<endl<<i+1<<" Customer Buy:"<<endl;
        for(int n1=0;n1<key_buy;n1++)
            cout<<ind_buy[n1]<<" ";
        cout<<endl<<"Not Buy:"<<endl;
        for(int n2=0;n2<key_not_buy;n2++)
            cout<<ind_not_buy[n2]<<" ";
        */

        float list_prod[prod_n];
        int list_prod_n=0, list_prod_n_ind=0, list_prod_ind[prod_n];

        for(int m=0;m<key_not_buy;m++)
        {
            float sum1=0;
            for(int l=0;l<key_buy;l++)
            {
                sum1=sum1+prod_prod[ind_buy[l]][ind_not_buy[m]];
            }

```

```

        list_prod[list_prod_n++]=sum1;
        list_prod_ind[list_prod_n_ind++]=ind_not_buy[m]+1;
    }

    /*
    //Unsorted List
    cout<<endl;
    for(int n1=0;n1<list_prod_n;n1++)
        cout<<list_prod[n1]<<": "<<list_prod_ind[n1]<<" ";
    cout<<endl;
    */

    //Sort non-purchased products according to the highest similarity first
    for(int m1 = 0; m1<list_prod_n; m1++)
    {
        for(int m2 = m1+1; m2<list_prod_n; m2++)
        {
            if(list_prod[m2] <= list_prod[m1])
            {
                float temp = list_prod[m1];
                list_prod[m1] = list_prod[m2];
                list_prod[m2] = temp;
                temp = list_prod_ind[m1];
                list_prod_ind[m1] = list_prod_ind[m2];
                list_prod_ind[m2] = temp;
            }
        }
    }
    /*
    //Sorted list
    cout<<endl;
    for(int n1=0;n1<list_prod_n;n1++)
        cout<<list_prod[n1]<<": "<<list_prod_ind[n1]<<" ";
    cout<<endl;
    */

    //Generate recommendations
    string s1;
    for(int m3=list_prod_n-1;m3>=0;m3--)
    {
        s1.append("P");
        s1.append(to_string(list_prod_ind[m3]));
        s1.append(" ");
    }
    //If recommendations list is empty
    if(s1=="")

```



```
        cout<<"C"<<i+1<<"-"<<"NO RECOMMENDATIONS"<<endl;
    else
        cout<<"C"<<i+1<<"-"<<s1<<endl;
    }
}
return 0;
}
```

TEST CASES

Sample Test Case 1	4 4 0 1 1 0 1 1 1 0 0 1 0 1 0 0 0 0 1 0.5 0 0.2 0.5 1 0.3 0.1 0 0.3 1 0.6 0.2 0.1 0.6 1	C1->P4 P1 C2->P4 C3->P3 P1 C4->P2 P3 P1 P4
Sample Test Case 2	4 4 0 1 1 0 1 1 1 1 0 1 0 1 0 0 0 0 1 0.5 0 0.2 0.5 1 0.3 0.1 0 0.3 1 0.6 0.2 0.1 0.6 1	C1->P4 P1 C2->NO RECOMMENDATIONS C3->P3 P1 C4->P2 P3 P4 P1
Hidden Test Case 1	1 1 0 1	C1->P1
Hidden Test Case 2	1 1 1 1	C1->NO RECOMMENDATIONS
Hidden Test Case 3	1 4 0 0 0 0 1 0.5 0.2 0.3 0.5 1 0.2 0.4 0.2 0.2 1 0.5 0.3 0.4 0.5 1	C1->P1 P2 P3 P4
Hidden Test Case 4	5 1 1 0 1 0 1 1	C1->NO RECOMMENDATIONS C2->P1 C3->NO RECOMMENDATIONS C4->P1 C5->NO RECOMMENDATIONS
Hidden Test Case 5	2 2 1 1 1 0 1 0.5	C1->NO RECOMMENDATIONS C2->P2

	0.5 1	
Hidden Test Case 6	3	C1->P1 P2 P3
	3	C2->NO RECOMMENDATIONS
	0 0 0	C3->P1 P2 P3
	1 1 1	
	0 0 0	
	1 0.5 0.2	
	0.5 1 0.3	
	0.2 0.3 1	

Hidden Test Case 7

5										C1->P10 P5 P7
10										P2 P3 P1
0	0	0	1	0	1	0	1	1	0	C2->P7 P9 P3 P2
1	0	0	1	1	1	0	1	0	1	C3->P6 P5 P10
1	0	1	1	0	0	0	1	0	0	P2 P7 P9
1	1	1	0	0	0	0	1	0	0	C4->P4 P5 P10
0	0	0	0	0	0	0	0	0	0	P6 P7 P9
1	0	0	0	0	0	0	0	0	0	C5->P8 P1 P4 P3
0	1	0.4	0.1	0.4	0.2	0	0.7	0.3	0.2	P6 P2 P5 P9 P10
0	0.4	1	0.5	0.3	0.5	0.2	0.3	0	0.1	P7
0	0.1	0.5	1	0.5	0.9	0.5	0.9	0.1	0.4	
0	0.4	0.3	0.5	1	0.8	0.4	0.6	0.2	0.6	
0	0.2	0.5	0.9	0.8	1	0.6	0.3	0.9	0.2	
0	0	0.2	0.5	0.4	0.6	1	0.5	0.5	0.5	
0	0.7	0.3	0.9	0.6	0.3	0.5	1	0.1	0.8	
0	0.3	0	0.1	0.2	0.9	0.5	0.1	1	0.7	
0	0.2	0.1	0.4	0.6	0.2	0.5	0.8	0.7	1	

Hidden Test Case 8

4										C1->P25 P1
25										C2->P25 P13
0	1	1	1	1	1	1	1	1	1	P24 P18 P5 P20
	1	1	1	1	1	1	1	1	1	P7 P10
1	1	1	1	0	1	0	1	1	0	C3->P2 P3 P6
	1	1	0	1	1	1	1	0	1	P11 P12 P14
	0	1	1	1	0	0				P15 P17 P19
0	0	0	0	0	0	0	0	0	0	P23 P1 P4 P7 P8
	0	0	0	0	0	0	0	0	0	P9 P10 P13 P16
1	1	1	0	0	1	1	0	0	1	P18 P21 P22
	1	1	1	1	1	0	1	1	1	P24 P5 P20 P25
	0	0	0	1	1	1				C4->P22 P20
1	0	0	0	0	0	0	0	0	0	P4 P8 P5 P16
	0	0	0	0	0	0	0	0	0	P21 P9
	0	0	0	0	0	0				
0	1	0.4	0.1	0.4	0.2	0	0.7	0.3	0.2	
	0	0.4	0.4	0.3	0.7	0.6	0.6	0.8	0.5	
	0.1	0.2	0.6	0	0.7	1				
0	0.4	1	0.5	0.3	0.5	0.2	0.3	0	0.1	
	0.8	0.3	0.7	0.7	0.3	0.6	1	0.7	0.3	
	0.2	0.5	0.4	0.9	0	1				
0	0.1	0.5	1	0.5	0.9	0.5	0.9	0.1	0.4	
	0.3	0.5	0.7	0.1	0.6	0.2	0.7	0.7	0.6	
	0.5	0.1	0.3	0.7	0.9	1				

0	0.4	0.3	0.5	1	0.8	0.4	0.6	0.2	0.6	
	0.1	0.4	0.1	0.2	1	0.4	0.8	1	0.3	
	0.1	0.8	0.7	0.2	1	1				
0	0.2	0.5	0.9	0.8	1	0.6	0.3	0.9	0.2	
	0	1	0.6	0.9	0.2	0.3	0.9	0.3	0.7	
	0.4	0.4	0.7	1	0.3	1				
0	0	0.2	0.5	0.4	0.6	1	0.5	0.5	0.5	
	0.4	0.2	0	0.9	0.6	0.6	0.3	0.7	0.9	
	0.8	0.2	0.3	0.3	0.7	1				
0	0.7	0.3	0.9	0.6	0.3	0.5	1	0.1	0.8	
	0.7	0.2	0.7	0.9	0.3	0.7	0	0.2	0.7	
	0.3	0.5	0.2	0.6	0.8	1				
0	0.3	0	0.1	0.2	0.9	0.5	0.1	1	0.7	
	0.2	0.2	0.8	0	0.7	0.4	0.3	0.3	0.9	
	0.3	0.7	0.1	0.6	0.2	1				
0	0.2	0.1	0.4	0.6	0.2	0.5	0.8	0.7	1	
	0.3	0.1	0.5	0.8	0.3	0.7	0.1	0.5	0.5	
	0.3	0.1	0.4	0.3	0	1				
0	0	0.8	0.3	0.1	0	0.4	0.7	0.2	0.3	
	1	0.2	1	0	0.6	0.5	0.8	0.7	0.5	
	1	0.6	0.9	0.8	0.7	1				
0	0.4	0.3	0.5	0.4	1	0.2	0.2	0.2	0.1	
	0.2	1	0.4	0.6	0.4	0.3	0.1	0.7	0.4	
	0.2	0.6	0.4	0.5	1	1				
0	0.4	0.7	0.7	0.1	0.6	0	0.7	0.8	0.5	
	1	0.4	1	0.6	0.6	0.4	0.6	0.4	0.7	
	0.6	0.9	0.6	0.4	0.7	1				
0	0.3	0.7	0.1	0.2	0.9	0.9	0.9	0	0.8	
	0	0.6	0.6	1	0.2	0.1	0.3	0.9	0.6	
	0.6	0.7	0.8	0.4	0.7	1				
0	0.7	0.3	0.6	1	0.2	0.6	0.3	0.7	0.3	
	0.6	0.4	0.6	0.2	1	0.4	0.9	0.1	0.4	
	0.9	0	0.8	0.3	0.8	1				
0	0.6	0.6	0.2	0.4	0.3	0.6	0.7	0.4	0.7	
	0.5	0.3	0.4	0.1	0.4	1	0.2	0.3	0.9	
	0.1	0.2	0.1	0.9	0.6	1				
0	0.6	1	0.7	0.8	0.9	0.3	0	0.3	0.1	
	0.8	0.1	0.6	0.3	0.9	0.2	1	0	0.5	
	0.4	0.8	0.7	0.2	0.4	1				
0	0.8	0.7	0.7	1	0.3	0.7	0.2	0.3	0.5	
	0.7	0.7	0.4	0.9	0.1	0.3	0	1	0.4	
	0.4	0.4	0.7	1	0.6	1				
0	0.5	0.3	0.6	0.3	0.7	0.9	0.7	0.9	0.5	
	0.5	0.4	0.7	0.6	0.4	0.9	0.5	0.4	10.1	
	0.8	0.7	0.5	0.1	0.2	1				
0	0.1	0.2	0.5	0.1	0.4	0.8	0.3	0.3	0.3	
	1	0.2	0.6	0.6	0.9	0.1	0.4	0.4	0.1	
	1	0.7	0.2	0.9	1	1				
0	0.2	0.5	0.1	0.8	0.4	0.2	0.5	0.7	0.1	
	0.6	0.6	0.9	0.7	0	0.2	0.8	0.4	0.8	
	0.7	1	0.7	0.2	0.6	1				
0	0.6	0.4	0.3	0.7	0.7	0.3	0.2	0.1	0.4	
	0.9	0.4	0.6	0.8	0.8	0.1	0.7	0.7	0.7	
	0.2	0.7	1	0.8	0.6	1				
0	0	0.9	0.7	0.2	1	0.3	0.6	0.6	0.3	
	0.8	0.5	0.4	0.4	0.3	0.9	0.2	1	0.5	
	0.9	0.2	0.8	1	0.1	1				
0	0.7	0	0.9	1	0.3	0.7	0.8	0.2	0	
	0.7	1	0.7	0.7	0.8	0.6	0.4	0.6	0.1	
	1	0.6	0.6	0.1	1	1				

1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1			

Hidden Test Case 9

[illegible]

1	1	1	1	0	1	0	1	1	C20->P22 P20 P4 P16 P5 P8 P21 P9
	0	1	1	0	1	1	1	1	
	0	1	0	1	1	1	0	0	C21->P1 P17 P8 P18 P14
0	0	0	1	0	0	0	0	0	P25 P5 P12 P21 P9 P10
	0	1	1	0	0	0	0	1	P2
	1	1	1	0	1	1	0	0	C22->P11 P4 P13 P19
1	1	1	0	0	1	1	0	0	P22 P12 P14 P17 P18
	1	1	1	1	1	1	0	1	P24 P8 P15 P21 P23 P2
	1	1	0	0	0	1	1	1	P16 P20 P1 P3 P6 P7 P9
0	0	1	1	0	1	1	0	0	P5 P10 P25
	0	1	0	1	0	1	1	0	C23->P24 P6 P1 P15 P23
	0	1	1	0	1	1	1	0	P19 P3 P8 P20 P7 P10 P2
0	0	0	0	0	0	0	0	0	P16
	0	0	0	0	0	0	0	0	C24->P25 P2
	0	0	0	0	0	0	0	0	C25->P1 P25
	0	0	1	1	0	0	0	1	
	0	1	1	1	1	0	0	1	
1	1	0	0	1	1	0	0	1	
	0	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1	0	
0	1	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1	0	
1	0.5	0.9	0.9	0.5	0.6	0.9	0	0.1	
	0.3	0.4	1	0.7	0.4	0.5	1	0.6	
	1	0.8	0.9	0.6	0.9	0.1	0.3	0.4	
0.5	1	0.4	0.1	0.4	0.2	0	0.7	0.3	
	0.2	0	0.4	0.4	0.3	0.7	0.6	0.6	
	0.8	0.5	0.1	0.2	0.6	0	0.7	0.7	
0.9	0.4	1	0.5	0.3	0.5	0.2	0.3	0	
	0.1	0.8	0.3	0.7	0.7	0.3	0.6	1	
	0.7	0.3	0.2	0.5	0.4	0.9	0	0.2	
0.9	0.1	0.5	1	0.5	0.9	0.5	0.9	0.1	
	0.4	0.3	0.5	0.7	0.1	0.6	0.2	0.7	
	0.7	0.6	0.5	0.1	0.3	0.7	0.9	0.5	
0.5	0.4	0.3	0.5	1	0.8	0.4	0.6	0.2	
	0.6	0.1	0.4	0.1	0.2	1	0.4	0.8	
	1	0.3	0.1	0.8	0.7	0.2	1	0.5	
0.6	0.2	0.5	0.9	0.8	1	0.6	0.3	0.9	
	0.2	0	1	0.6	0.9	0.2	0.3	0.9	
	0.3	0.7	0.4	0.4	0.7	1	0.3	0.3	
0.9	0	0.2	0.5	0.4	0.6	1	0.5	0.5	
	0.5	0.4	0.2	0	0.9	0.6	0.6	0.3	
	0.7	0.9	0.8	0.2	0.3	0.3	0.7	0.7	
0	0.7	0.3	0.9	0.6	0.3	0.5	1	0.1	
	0.8	0.7	0.2	0.7	0.9	0.3	0.7	0	
	0.2	0.7	0.3	0.5	0.2	0.6	0.8	0.6	
0.1	0.3	0	0.1	0.2	0.9	0.5	0.1	1	
	0.7	0.2	0.2	0.8	0	0.7	0.4	0.3	
	0.3	0.9	0.3	0.7	0.1	0.6	0.2	0.2	
0.3	0.2	0.1	0.4	0.6	0.2	0.5	0.8	0.7	
	1	0.3	0.1	0.5	0.8	0.3	0.7	0.1	
	0.5	0.5	0.3	0.1	0.4	0.3	0	0.3	
0.4	0	0.8	0.3	0.1	0	0.4	0.7	0.2	
	0.3	1	0.2	1	0	0.6	0.5	0.8	
	0.7	0.5	1	0.6	0.9	0.8	0.7	0.3	
1	0.4	0.3	0.5	0.4	1	0.2	0.2	0.2	
	0.1	0.2	1	0.4	0.6	0.4	0.3	0.1	
	0.7	0.4	0.2	0.6	0.4	0.5	1	0.9	

0.7	0.4	0.7	0.7	0.1	0.6	0	0.7	0.8	
	0.5	1	0.4	1	0.6	0.6	0.4	0.6	
	0.4	0.7	0.6	0.9	0.6	0.4	0.7	0.4	
0.4	0.3	0.7	0.1	0.2	0.9	0.9	0.9	0	
	0.8	0	0.6	0.6	1	0.2	0.1	0.3	
	0.9	0.6	0.6	0.7	0.8	0.4	0.7	0.9	
0.5	0.7	0.3	0.6	1	0.2	0.6	0.3	0.7	
	0.3	0.6	0.4	0.6	0.2	1	0.4	0.9	
	0.1	0.4	0.9	0	0.8	0.3	0.8	0.8	
1	0.6	0.6	0.2	0.4	0.3	0.6	0.7	0.4	
	0.7	0.5	0.3	0.4	0.1	0.4	1	0.2	
	0.3	0.9	0.1	0.2	0.1	0.9	0.6	0.8	
0.6	0.6	1	0.7	0.8	0.9	0.3	0	0.3	
	0.1	0.8	0.1	0.6	0.3	0.9	0.2	1	
	0	0.5	0.4	0.8	0.7	0.2	0.4	0.6	
1	0.8	0.7	0.7	1	0.3	0.7	0.2	0.3	
	0.5	0.7	0.7	0.4	0.9	0.1	0.3	0	
	1	0.4	0.4	0.4	0.7	1	0.6	0.7	
0.8	0.5	0.3	0.6	0.3	0.7	0.9	0.7	0.9	
	0.5	0.5	0.4	0.7	0.6	0.4	0.9	0.5	
	0.4	10.1	0.8	0.7	0.5	0.1	0.2	0.2	
0.9	0.1	0.2	0.5	0.1	0.4	0.8	0.3	0.3	
	0.3	1	0.2	0.6	0.6	0.9	0.1	0.4	
	0.4	0.1	1	0.7	0.2	0.9	1	0.4	
0.6	0.2	0.5	0.1	0.8	0.4	0.2	0.5	0.7	
	0.1	0.6	0.6	0.9	0.7	0	0.2	0.8	
	0.4	0.8	0.7	1	0.7	0.2	0.6	0.1	
0.9	0.6	0.4	0.3	0.7	0.7	0.3	0.2	0.1	
	0.4	0.9	0.4	0.6	0.8	0.8	0.1	0.7	
	0.7	0.7	0.2	0.7	1	0.8	0.6	0.3	
0.1	0	0.9	0.7	0.2	1	0.3	0.6	0.6	
	0.3	0.8	0.5	0.4	0.4	0.3	0.9	0.2	
	1	0.5	0.9	0.2	0.8	1	0.1	0.8	
0.3	0.7	0	0.9	1	0.3	0.7	0.8	0.2	
	0	0.7	1	0.7	0.7	0.8	0.6	0.4	
	0.6	0.1	1	0.6	0.6	0.1	1	0.6	
0.4	0.7	0.2	0.5	0.5	0.3	0.7	0.6	0.2	
	0.3	0.3	0.9	0.4	0.9	0.8	0.8	0.6	
	0.7	0.2	0.4	0.1	0.3	0.8	0.6	1	

Hidden Test Case 10

25										C1->P13 P19 P24
25										P3 P22 P15 P1 P7
0	0	0	1	0	1	0	1	1	0	P5 P18 P14 P25
	1	0	0	0	0	0	1	0	0	P12 P16 P10 P2
	1	1	0	1	0	0				
1	0	0	1	1	1	0	1	0	1	C2->P14 P7 P12
	1	0	1	0	1	0	0	1	1	P16 P17 P21 P3
	1	0	1	1	1	1				P2 P9
1	0	1	1	0	0	0	1	0	0	C3->P14 P11 P12
	0	0	1	0	0	1	0	1	1	P5 P20 P7 P23 P2
	0	1	0	0	1	1				P17 P22 P6 P15
1	1	1	0	0	0	0	1	0	0	P10 P9
	1	0	1	0	0	1	1	1	0	
	1	1	0	0	1	0				C4->P22 P4 P19
0	1	0	1	0	1	1	1	0	1	P14 P15 P23 P5
	0	1	0	0	0	0	1	1	0	P25 P12 P7 P6
	1	0	1	1	1	0				P10 P9

1	0	0	1	0	0	1	0	1	0	C5->P14 P1 P5
	1	0	1	1	1	0	0	1	1	P25 P15 P11 P13
	0	0	1	0	0	1				P19 P16 P3 P21
0	0	0	0	1	1	0	0	0	1	P9
	0	1	1	1	0	0	0	0	1	C6->P20 P24 P6
	1	0	1	1	1	0				P23 P17 P12 P8
0	0	0	0	0	0	1	1	0	0	P3 P16 P21 P10
	1	0	1	1	1	0	0	1	1	P5 P2
	0	0	1	0	1	0				C7->P18 P1 P21
0	1	0	1	0	0	1	1	0	1	P4 P8 P15 P7 P25
	0	1	1	0	1	1	1	0	1	P11 P17 P9 P16
	1	0	0	0	0	0				P3 P2
	0	1	1	1	0	0				C8->P20 P1 P4
	1	1	1	1	1	0	0	0	0	P25 P5 P21 P23
	1	1	1	0	1	1	1	1	1	P2 P6 P10 P16
1	1	0	0	0	0	0				P12 P17 P3 P9
	1	1	1	1	0	1	0	0	1	C9->P1 P24 P25
	1	1	0	1	1	1	1			P6 P11 P14 P5 P9
1	1	0	1	1	0	0	0	0	0	P18 P23 P22 P21
	0	0	1	0	1	1	1	1	0	P3
1	1	0	1	1	1	1	1	1	1	C10->P1 P19 P24
	0	0	1	1	1	0	0	0	1	P23 P18 P25 P15
	1	1	0	0	1	0				P5 P20 P10 P9
0	0	1	1	1	0	1	0	0	0	C11->P1 P24 P22
	0	0	0	1	0	0	1	1	1	P25 P23 P3 P21
	0	1	1	1	1	1				P8 P7 P9 P14 P10
1	1	1	1	0	1	1	0	1	0	C12->P13 P22 P6
	1	1	1	1	1	1	0	0	1	P17 P16 P14 P8
	0	1	1	0	1	0	1	0	0	P10
	1	1	0	1	1	1	1	1	0	C13->P18 P6 P11
1	1	1	0	1	0	1	0	1	0	P8 P7 P12 P3 P21
	1	1	1	0	1	1	1	0	1	P14 P9 P10
	0	1	1	1	1	0	0			C14->P7 P22 P5
0	0	0	1	0	0	0	0	0	0	P11 P4 P16 P25
	1	1	0	0	0	0	0	1	1	P12 P6 P17 P3
	1	0	1	1	0	0				P23
1	1	1	0	0	1	1	0	0	1	C15->P1 P6 P20
	1	1	1	1	1	0	1	1	1	P11 P15 P8 P13
0	0	1	1	0	1	1	0	0	0	P12 P16 P2 P10
	1	0	1	0	1	1	0	0	1	P9
	1	0	1	1	1	0				C16->P17 P18
0	0	0	0	0	0	1	0	0	0	P20 P8 P25 P5
	1	1	1	1	1	0	0	1	0	P23 P10
	0	0	1	1	0	0	0	0	1	C17->P24 P22 P3
	1	1	1	1	0	0	1	1	0	P6 P19 P13 P4 P8
	0	1	1	0	0	1				P10 P9
1	0	1	1	1	1	0	1	1	1	C18->P13 P18
	1	0	0	0	1	0	1	0	0	P24 P25 P20 P5
	0	0	1	1	0	1	1	0	1	P7 P10
0	1	0	1	0	0	0	0	0	1	
	0	0	0	0	0	0	0	1	1	
	1	1	1	0	1	1	1			

1	0.5	0.9	0.9	0.5	0.6	0.9	0	0.1	0.3	C19->P1 P6 P24 P13 P3 P15 P21 P25 P7 P14 P5 P8 P16 P2 P9 P10 C20->P22 P20 P4 P16 P5 P8 P21 P9 C21->P1 P17 P8 P18 P14 P25 P5 P12 P21 P9 P10 P2 C22->P1 P24 P25 P4 P3 P19 P7 P21 P8 P5 P17 P9 P16 P2 C23->P24 P6 P1 P15 P23 P19 P3 P8 P20 P7 P10 P2 P16 C24->P13 P18 P19 P16 P20 P7 P24 P14 P12 P21 P2 C25->P14 P1 P5 P13 P11 P15 P23 P16 P12 P6 P17 P9 P3
	0.4	1	0.7	0.4	0.5	1	0.6	1	0.8	
	0.9	0.6	0.9	0.1	0.3	0.4				
0.5	1	0.4	0.1	0.4	0.2	0	0.7	0.3	0.2	
	0	0.4	0.4	0.3	0.7	0.6	0.6	0.8	0.5	
	0.1	0.2	0.6	0	0.7	0.7				
0.9	0.4	1	0.5	0.3	0.5	0.2	0.3	0	0.1	
	0.8	0.3	0.7	0.7	0.3	0.6	1	0.7	0.3	
	0.2	0.5	0.4	0.9	0	0.2				
0.9	0.1	0.5	1	0.5	0.9	0.5	0.9	0.1	0.4	
	0.3	0.5	0.7	0.1	0.6	0.2	0.7	0.7	0.6	
	0.5	0.1	0.3	0.7	0.9	0.5				
0.5	0.4	0.3	0.5	1	0.8	0.4	0.6	0.2	0.6	
	0.1	0.4	0.1	0.2	1	0.4	0.8	1	0.3	
	0.1	0.8	0.7	0.2	1	0.5				
0.6	0.2	0.5	0.9	0.8	1	0.6	0.3	0.9	0.2	
	0	1	0.6	0.9	0.2	0.3	0.9	0.3	0.7	
	0.4	0.4	0.7	1	0.3	0.3				
0.9	0	0.2	0.5	0.4	0.6	1	0.5	0.5	0.5	
	0.4	0.2	0	0.9	0.6	0.6	0.3	0.7	0.9	
	0.8	0.2	0.3	0.3	0.7	0.7				
0	0.7	0.3	0.9	0.6	0.3	0.5	1	0.1	0.8	
	0.7	0.2	0.7	0.9	0.3	0.7	0	0.2	0.7	
	0.3	0.5	0.2	0.6	0.8	0.6				
0.1	0.3	0	0.1	0.2	0.9	0.5	0.1	1	0.7	
	0.2	0.2	0.8	0	0.7	0.4	0.3	0.3	0.9	
	0.3	0.7	0.1	0.6	0.2	0.2				
0.3	0.2	0.1	0.4	0.6	0.2	0.5	0.8	0.7	1	
	0.3	0.1	0.5	0.8	0.3	0.7	0.1	0.5	0.5	
	0.3	0.1	0.4	0.3	0	0.3				
0.4	0	0.8	0.3	0.1	0	0.4	0.7	0.2	0.3	
	1	0.2	1	0	0.6	0.5	0.8	0.7	0.5	
	1	0.6	0.9	0.8	0.7	0.3				
1	0.4	0.3	0.5	0.4	1	0.2	0.2	0.2	0.1	
	0.2	1	0.4	0.6	0.4	0.3	0.1	0.7	0.4	
	0.2	0.6	0.4	0.5	1	0.9				
0.7	0.4	0.7	0.7	0.1	0.6	0	0.7	0.8	0.5	
	1	0.4	1	0.6	0.6	0.4	0.6	0.4	0.7	
	0.6	0.9	0.6	0.4	0.7	0.4				
0.4	0.3	0.7	0.1	0.2	0.9	0.9	0.9	0	0.8	
	0	0.6	0.6	1	0.2	0.1	0.3	0.9	0.6	
	0.6	0.7	0.8	0.4	0.7	0.9				
0.5	0.7	0.3	0.6	1	0.2	0.6	0.3	0.7	0.3	
	0.6	0.4	0.6	0.2	1	0.4	0.9	0.1	0.4	
	0.9	0	0.8	0.3	0.8	0.8				
1	0.6	0.6	0.2	0.4	0.3	0.6	0.7	0.4	0.7	
	0.5	0.3	0.4	0.1	0.4	1	0.2	0.3	0.9	
	0.1	0.2	0.1	0.9	0.6	0.8				
0.6	0.6	1	0.7	0.8	0.9	0.3	0	0.3	0.1	
	0.8	0.1	0.6	0.3	0.9	0.2	1	0	0.5	
	0.4	0.8	0.7	0.2	0.4	0.6				
1	0.8	0.7	0.7	1	0.3	0.7	0.2	0.3	0.5	
	0.7	0.7	0.4	0.9	0.1	0.3	0	1	0.4	
	0.4	0.4	0.7	1	0.6	0.7				
0.8	0.5	0.3	0.6	0.3	0.7	0.9	0.7	0.9	0.5	
	0.5	0.4	0.7	0.6	0.4	0.9	0.5	0.4	10.1	
	0.8	0.7	0.5	0.1	0.2	0.2				
0.9	0.1	0.2	0.5	0.1	0.4	0.8	0.3	0.3	0.3	
	1	0.2	0.6	0.6	0.9	0.1	0.4	0.4	0.1	
	1	0.7	0.2	0.9	1	0.4				

0.6	0.2	0.5	0.1	0.8	0.4	0.2	0.5	0.7	0.1	
	0.6	0.6	0.9	0.7	0	0.2	0.8	0.4	0.8	
	0.7	1	0.7	0.2	0.6	0.1				
0.9	0.6	0.4	0.3	0.7	0.7	0.3	0.2	0.1	0.4	
	0.9	0.4	0.6	0.8	0.8	0.1	0.7	0.7	0.7	
	0.2	0.7	1	0.8	0.6	0.3				
0.1	0	0.9	0.7	0.2	1	0.3	0.6	0.6	0.3	
	0.8	0.5	0.4	0.4	0.3	0.9	0.2	1	0.5	
	0.9	0.2	0.8	1	0.1	0.8				
0.3	0.7	0	0.9	1	0.3	0.7	0.8	0.2	0	
	0.7	1	0.7	0.7	0.8	0.6	0.4	0.6	0.1	
	1	0.6	0.6	0.1	1	0.6				
0.4	0.7	0.2	0.5	0.5	0.3	0.7	0.6	0.2	0.3	
	0.3	0.9	0.4	0.9	0.8	0.8	0.6	0.7	0.2	
	0.4	0.1	0.3	0.8	0.6	1				