Terna Engineering College
Computer Engineering Department

Program: Sem VII

Course: Artificial Intelligence & Soft Computing (AI&SC)

Experiment No. 06

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	Name: AMEY THAKUR
Class: BE-COMPS-50	Batch: B3
Date of Experiment: 21-09-2021	Date of Submission: 21-09-2021
Grade :	

Aim: To implement a program to calculate cartesian product on fuzzy relation using

1. Max-Min and
2. Max-Product composition.

B.1 Software Code written by a student:

(Paste your Problem Statement for Classification and Data set Used as a knowledge Database for Given Classification Problem)

```
#include<stdio.h>

void main()

{

    int i,j,m1,n1,m2,n2,c,k;

    float R[50][50],S[50][50],T[50][50],U[50][50],a[50];

    printf("Enter the number of rows and columns of matrix R: ");

    scanf("%d %d",&m1,&n1);

    printf("Enter the Elements:\n");

    for(i=0;i<m1;i++)
```

```
{
        for(j=0;j<n1;j++)
    {
        scanf("%f", &R[i][j]);
    }
}
printf("\nEnter the number of rows and columns of matrix S: ");
scanf("%d %d",&m2,&n2);
printf("Enter the Elements:\n");
for(i=0;i<m2;i++)
{
        for(j=0;j<n2;j++)
    {
        scanf("%f", &S[i][j]);
    }
}
if(n1!=m2)
{
    printf("Cartesian product can't be determined.");
}
printf("\nCartesian Product on Fuzzy Relation using:\n\n1. Max-Min Composition\n2. Max-Product Composition\n Enter your choice:");
scanf("%d",&c);
switch(c)
{
    case 1:
        for(i=0;i<m1;i++)
```

```
                {
                        for(j=0;j<n2;j++)
                        {
                                for(k=0;k<m2;k++)
                                {
                                        if(R[i][k]>=S[k][j])
                                        {
                                                a[k]=S[k][j];
                                        }
                                        else if(R[i][k]<=S[k][j])
                                        {
                                                a[k]=R[i][k];
                                        }
                                }
                                for(k=1;k<m2;k++)
                                {
                                        if(a[0]<a[k])
                                                a[0]=a[k];
                                }
                                T[i][j]=a[0];
                        }
}
printf("\nThe Max-Min of R and S is T:\n");
for(i=0;i<m1;i++)
{
        for(j=0;j<n2;j++)
        {
```

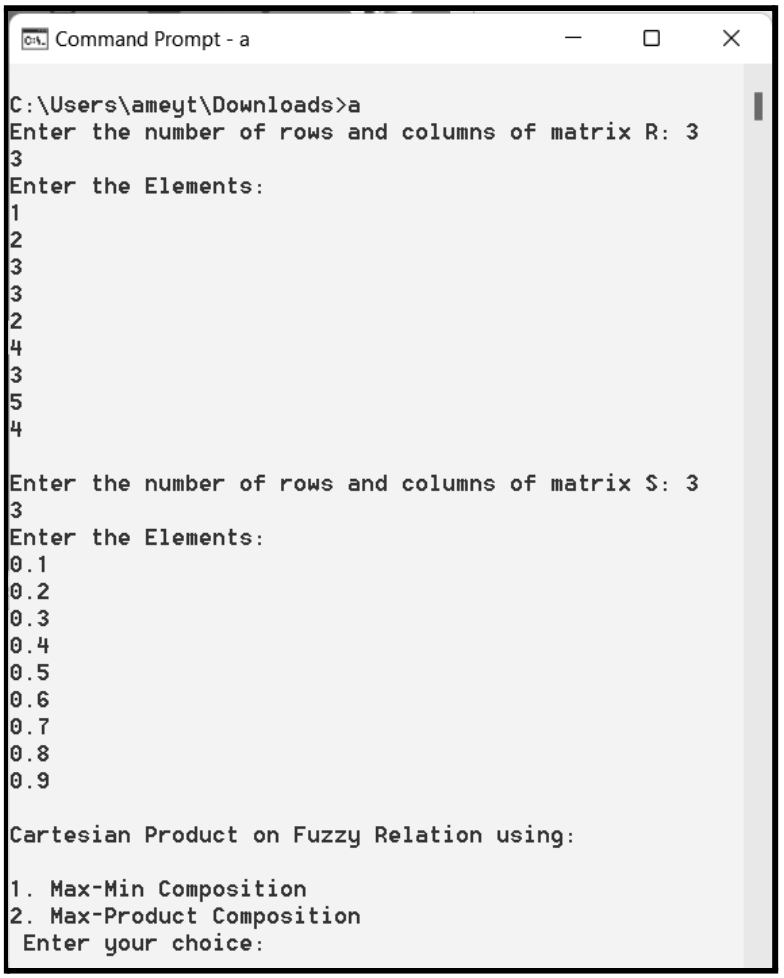
```
                printf(" %f",T[i][j]);
        }
        printf("\n");
}
        break;
case 2:
        for(i=0;i<m1;i++)
        {
                for(j=0;j<n2;j++)
                        {
                                for(k=0;k<m2;k++)
                                {
                                        a[k]=R[i][k]*S[k][j];
                                }
                                for(k=1;k<m2;k++)
                                {
                                        if(a[0]<a[k])
                                        a[0]=a[k];
                                }
                                U[i][j]=a[0];
                        }
}
printf("\nThe Max-Product of R and S is U:\n");
for(i=0;i<m1;i++)
{
        for(j=0;j<n2;j++)
        {
```

```
                printf(" %f",U[i][j]);
            }
            printf("\n");
        }
        break;
        default:
        printf("Please choose correct option.");
        }
}
```

B.2 Input and Output:

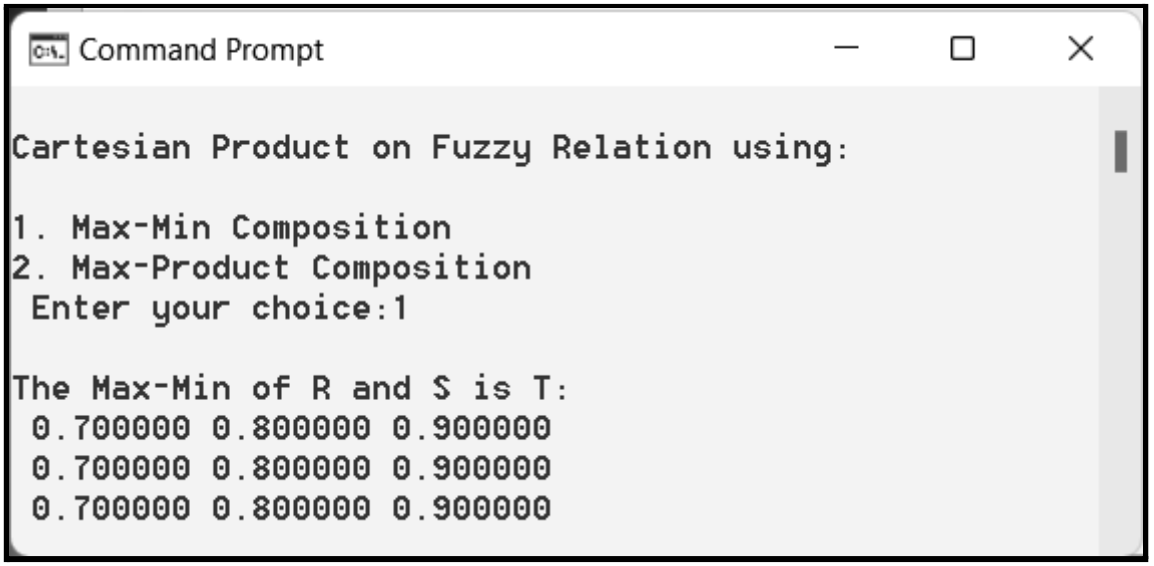
(Paste your screenshot of Analysis of Data, Relevant Attributes Selection by using at least three methods)

Input:

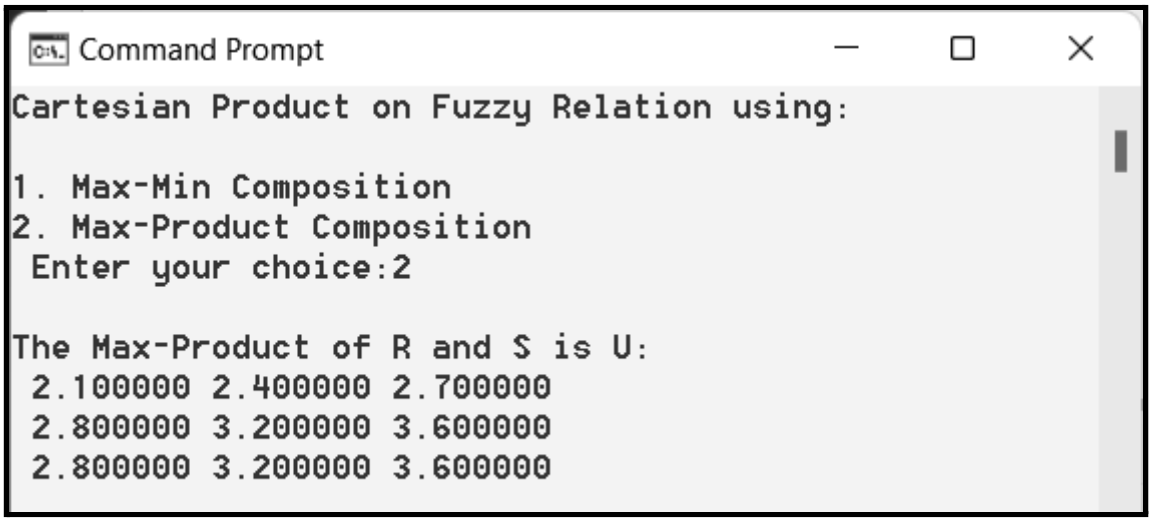


Output:

Max-Min Composition:



Max-Product composition:



B.3 Observations and learning: (Performance Evaluation)

(Students are expected to comment on the output obtained with clear observations getting from Performance Evaluation after analyzing the data and learning for each task assigned)

As a result, utilising 1. Max-Min and 2. Max-Product composition, we were able to effectively develop a programme to compute cartesian product on fuzzy relations.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

Hence we've successfully implemented a program to calculate cartesian product on fuzzy relation using 1. Max-Min and 2. Max-Product composition.

B.5 Question of Curiosity

(To be answered by student based on the practical performed and learning/observations)

Q1)

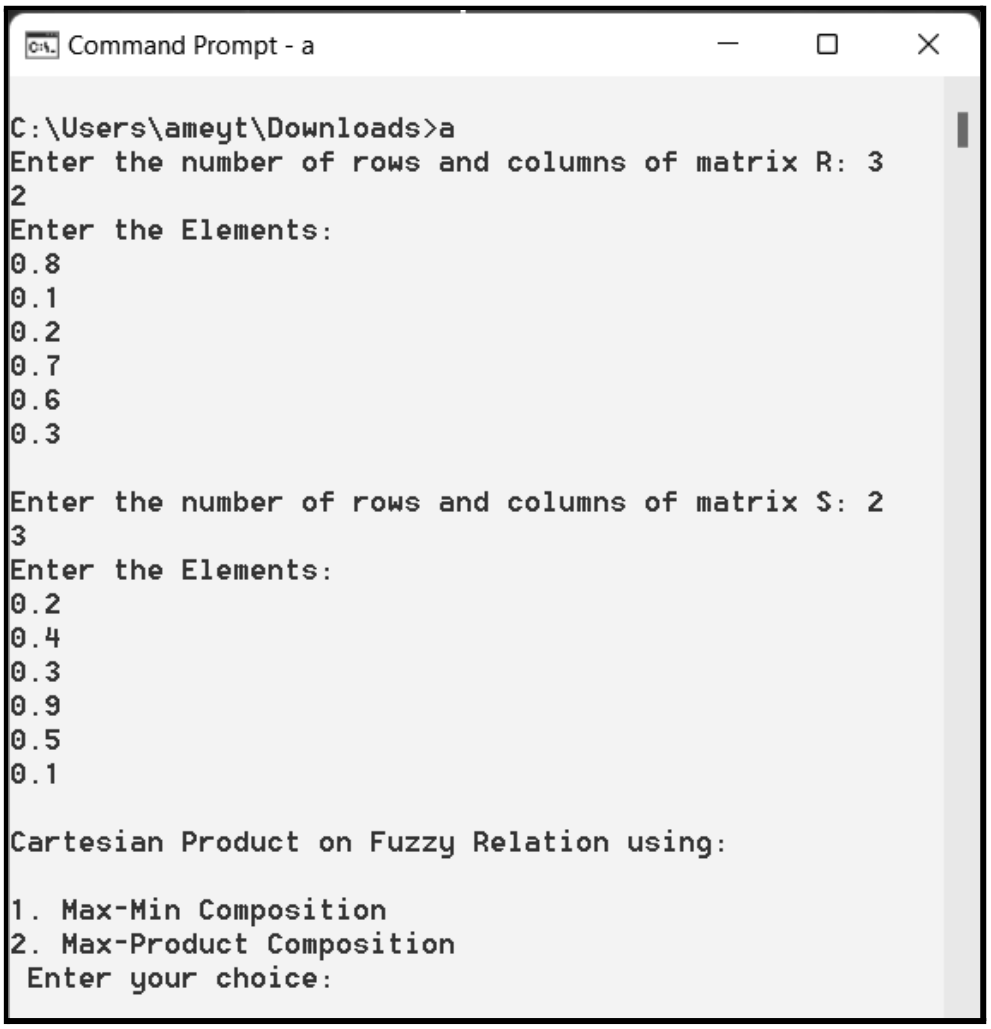
Consider R and S be fuzzy relations defined as follows

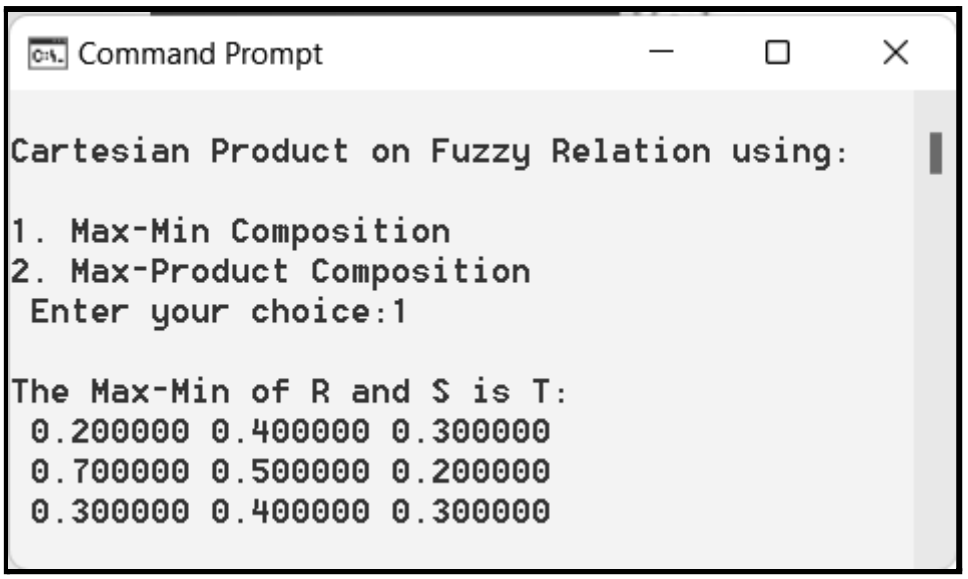
$$\tilde{R} = \begin{bmatrix} 0.8 & 0.1 \\ 0.2 & 0.7 \\ 0.6 & 0.3 \end{bmatrix} \quad \tilde{S} = \begin{bmatrix} 0.2 & 0.4 & 0.3 \\ 0.9 & 0.5 & 0.1 \end{bmatrix}$$

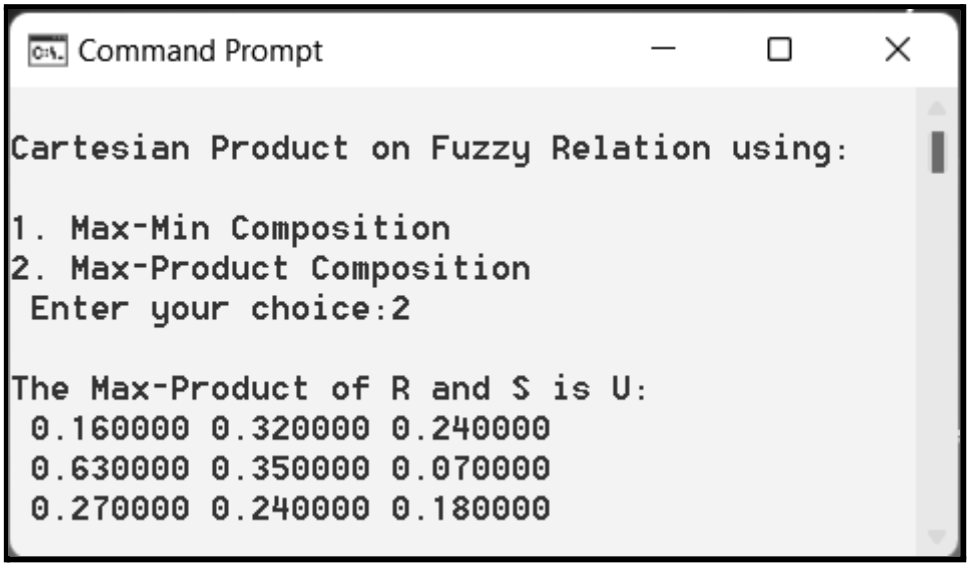
Find the following fuzzy compositions

1. $\tilde{T} = \tilde{R} \circ \tilde{S}$
2. $\tilde{U} = \tilde{R} \bullet \tilde{S}$

Ans:




Command Prompt
Cartesian Product on Fuzzy Relation using:
1. Max-Min Composition
2. Max-Product Composition
Enter your choice:1
The Max-Min of R and S is T:
0.200000 0.400000 0.300000
0.700000 0.500000 0.200000
0.300000 0.400000 0.300000


Command Prompt
Cartesian Product on Fuzzy Relation using:
1. Max-Min Composition
2. Max-Product Composition
Enter your choice:2
The Max-Product of R and S is U:
0.160000 0.320000 0.240000
0.630000 0.350000 0.070000
0.270000 0.240000 0.180000