```
/*
    Є кілька множин.
    Необхідно перевірити такі умови: чи збігаються ці множини; перетинаються чи
не перетинаються?
    Результати подайте у вигляді таблиці (матриці).
*/

#include <iostream>
#include <string>
#include <iomanip>
#include <algorithm>

using namespace std;


int main()
{
    int pluralCount;

    cout << "Input plural amount: ";
    cin >> pluralCount;

    double** plurals = new double*[pluralCount];
    double** pluralsSorted = new double*[pluralCount];
    int* pluralsSize = new int[pluralCount];
    string **pluralCoincide = new string*[pluralCount+1];
    string **pluralIntersecting = new string*[pluralCount+1];

    for (int i = 0; i < pluralCount+1; i++)
    {
        pluralCoincide[i] = new string[pluralCount+1];
        pluralCoincide[i][0] = i;
        pluralCoincide[0][i] = i;

        pluralIntersecting[i] = new string[pluralCount+1];
        pluralIntersecting[i][0] = i;
        pluralIntersecting[0][i] = i;
    }

    pluralCoincide[0][0] = "Plurals";
    pluralIntersecting[0][0] = "Plurals";

    for (int i = 0; i < pluralCount; i++)
    {
        string plural;

        cout << "Input plural number " << i + 1 << " size: ";
        cin >> pluralsSize[i];

        plurals[i] = new double[pluralsSize[i]];
        pluralsSorted[i] = new double[pluralsSize[i]];
```

```
        cout << "Input plural in line through a space: ";

        for (int j = 0; j < pluralsSize[i]; j++)
        {
            cin >> plurals[i][j];
            pluralsSorted[i][j] = plurals[i][j];
        }

        sort(pluralsSorted[i], pluralsSorted[i]+pluralsSize[i]);
    }

    cout << "Convergence of sets: " << endl;

    for (int i = 0; i < pluralCount; i++)
    {

        for (int j = 0; j < pluralCount; j++)
        {
            if (pluralsSize[i] == pluralsSize[j])
            {
                if (i == j)
                {
                    pluralCoincide[i+1][j+1] = "true";
                }
                else
                {
                    for (int x = 0; x < pluralsSize[i]; x++)
                    {
                        if (pluralsSorted[i][x] != pluralsSorted[j][x])
                        {
                            pluralCoincide[i+1][j+1] = "false";
                            break;
                        }
                    }

                    if (pluralCoincide[i+1][j+1] != "false")
                    {
                        pluralCoincide[i+1][j+1] = "true";
                    }
                }
            }
            else
            {
                pluralCoincide[i+1][j+1] = "false";
            }
        }
    }

    for (int i = 0; i < pluralCount+1; i++)
    {
```

```
        for (int j = 0; j < pluralCount+1; j++)
        {
            cout << setw(8) << pluralCoincide[i][j] << " ";
        }
        cout << endl;
    }

    cout << "Intersecting sets: " << endl;

    for (int i = 0; i < pluralCount; i++)
    {

        for (int j = 0; j < pluralCount; j++)
        {
            if (pluralCoincide[i+1][j+1] != "true")
            {
               for (int x = 0; x < pluralsSize[i]; x++)
               {
                    for (int y = 0; y < pluralsSize[j]; y++)
                    {
                        if (pluralsSorted[i][x] == pluralsSorted[j][y])
                        {
                            pluralIntersecting[i+1][j+1] = "true";
                            break;
                        }
                    }
                    if (pluralIntersecting[i+1][j+1] == "true")
                    {
                        break;
                    }
               }

                if (pluralIntersecting[i+1][j+1] != "true")
                {
                    pluralIntersecting[i+1][j+1] = "false";
                }

            }
            else
            {
                pluralIntersecting[i+1][j+1] = "true";
            }
        }
    }

    for (int i = 0; i < pluralCount+1; i++)
    {
        for (int j = 0; j < pluralCount+1; j++)
        {
            cout << setw(8) << pluralIntersecting[i][j] << " ";
        }
```

```
        cout << endl;
    }
}
```

```
Input plural amount: 3
Input plural number 1 size: 3
Input plural in line through a space: 1 2 3
Input plural number 2 size: 3
Input plural in line through a space: 3 2 1
Input plural number 3 size: 3
Input plural in line through a space: 2 3 1
Convergence of sets:
 Plurals        ⊖        ☻        ♥
       ⊖     true     true     true
       ☻     true     true     true
       ♥     true     true     true
Intersecting sets:
 Plurals        ⊖        ☻        ♥
       ⊖     true     true     true
       ☻     true     true     true
       ♥     true     true     true
PS D:\Денис Чумпер\Дискретні Структури\Лабор
```

```
Input plural amount: 4
Input plural number 1 size: 3
Input plural in line through a space: 1 2 3
Input plural number 2 size: 2
Input plural in line through a space: 2 2
Input plural number 3 size: 4
Input plural in line through a space: 1 2 3 4
Input plural number 4 size: 4
Input plural in line through a space: 4 3 2 1
Convergence of sets:
 Plurals        ⊖        ☻        ♥        ♦
       ⊖     true    false    false    false
       ☻    false     true    false    false
       ♥    false    false     true     true
       ♦    false    false     true     true
Intersecting sets:
 Plurals        ⊖        ☻        ♥        ♦
       ⊖     true     true     true     true
       ☻     true     true     true     true
       ♥     true     true     true     true
       ♦     true     true     true     true
```

```
Input plural amount: 3
Input plural number 1 size: 2
Input plural in line through a space: 1 2
Input plural number 2 size: 3
Input plural in line through a space: 3 4 5
Input plural number 3 size: 1
Input plural in line through a space: 2
Convergence of sets:
 Plurals        ☻        ☻        ♥
       ☻     true    false    false
       ☻    false     true    false
       ♥    false    false     true
Intersecting sets:
 Plurals        ☻        ☻        ♥
       ☻     true    false     true
       ☻    false     true    false
       ♥     true    false     true
```