

Magic squares. An $n \times n$ matrix that is filled with the numbers 1, 2, 3, . . . , n^2 is a magic square if the sum of the elements in each row, in each column, and in the two diagonals is the same value.

Write a program that reads in 16 values from the keyboard and tests whether they form a magic square when put into a 4×4 array. You need to test two features:

1. Does each of the numbers 1, 2, ..., 16 occur in the user input?
2. When the numbers are put into a square, are the sums of the rows, columns, and diagonals equal to each other?

Answer:

```
#include "stdafx.h"
#include <iostream>
using namespace std;

bool EnterTheArray(int a[])
{
    for (int i = 1; i < 17; i++)
    {
        bool b = false;
        for (int j = 0; j < 16; j++)
        {
            if (a[j] == i)
            {
                b = true;
            }
        }
        if (!b) return false;
    }
}

bool MagicSquares(int a[])
{
    int sum = a[0] + a[5] + a[10] + a[15];
    int newSum = a[3] + a[6] + a[9] + a[12];
    if (sum != newSum) return 0;
    for (int i = 0; i < 4; i++)
    {
        newSum = 0;
        for (int j = 0; j < 4; j++)
        {
            newSum += a[j + 4*i];
        }
        if (sum != newSum) return 0;
    }

    for (int i = 0; i < 4; i++)
    {
        newSum = 0;
        for (int j = 0; j < 4; j++)
        {
            newSum += a[4*j + i];
        }
        if (sum != newSum) return 0;
    }
    return true;
}

void Write(bool b)
{

```

```

        if (b) cout << "True" << endl;
        else cout << "False" << endl;
    }

int main()
{
    int a[16];
    cout << "Enter 16 numbers:";
    for (int i = 0; i<16; i++)
    {
        cin >> a[i];
    }
    cout << "1.Does each of the numbers 1, 2, ..., 16 occur in the user input?" << endl;
    Write(EnterTheArray(a));
    cout << "2.When the numbers are put into a square, are the sums of the rows, columns,
and diagonals equal to each other? " << endl;
    Write(MagicSquares(a));

    cout << "Press any key and Enter" << endl;

    cin >> a[1];
    return 0;
}

```