

```

using System;
using System.Text;
using System.Numerics;
using real = System.Double;
//using real = System.Single;
using static System.Math;
using static System.Numerics.Complex;
using static System.Console;
using static L2.Tools;

namespace L2
{
    class Table<T>
    {
        T[][] data;
        public Table(params T[][] data)
        { this.data = data; }
        public override string ToString()
        {
            StringBuilder sb = new StringBuilder();
            int ncols = data.Length;
            for (int irow = 0; irow < data[0].Length; irow++)
            {
                for (int icol = 0; icol < ncols; icol++)
                    sb.AppendFormat("{0} ", data[icol][irow]);
                sb.AppendLine();
            }
            return sb.ToString();
        }
    }
    static class Tools

```

```

{
    public static real[] LinSpace(real a, real b, int n = 100)
    {
        real[] t = new real[n + 1];
        real dt = (b - a) / n;
        for (int i = 0; i < n; i++)
        {
            t[i] = a;
            a += dt;
        }
        t[n] = b;
        return t;
    }

    public static T2[] Vectorize<T1,T2>(Func<T1, T2> f, T1[] x)
    {
        int n = x.Length;
        T2[] y = new T2[n];
        for (int i = 0; i < n; i++)
        {
            y[i] = f(x[i]);
        }
        return y;
    }
}

class Program
{
    static double[] r = { 1, 0.5, 0.3 };
    static double[] omega = { 1, 3, 8 };
}

```

```

static Complex dft(real t)
{
    Complex z = Zero;
    Complex j = ImaginaryOne;
    int n = r.Length;
    for (int i = 0; i < n; i++)
    {
        z += r[i] * Exp(j * omega[i] * t);
    }
    return z;
}
static real Real(Complex z)
{
    return z.Real;
}
static real Imaginary(Complex z)
{
    return z.Imaginary;
}
static void Main(string[] args)
{
    real[] t = LinSpace(0, 2*PI, 300);
    Complex[] cft = Vectorize(dft, t);
    real[][] xy = new real[2][];
    xy[0] = Vectorize(Real, cft);
    xy[1] = Vectorize(Imaginary, cft);
    Table<real> tab = new Table<real>(xy);
    WriteLine(tab);
}
}

```

}

