Matric's determinant and the Cramer method to solve linear equations

Programs in Dev C++ / Borland TurboC++

The topics covering two programs presented here are purely academic, but at the same time not as simple as all other operations on matrices.

1. **Calculation of a determinant of a quadratic matrix - max 10x10 of matrix dimensions**

   Possible expansion to a larger square matrix - the 'classical' method.
   To check the correctness of the results of the determinant's calculations, we can use more simple calculation methods, such as the one using the fact that if there are only zeros below (or above) the diagonal (the matrix should be brought to this state), then the determinant of the matrix is equal to the product of numbers on the diagonal.

2. **Solving a system of 'n' linear equations with 'n' variables - up to 10 equations with 10 variables - Cramer method.**

   Possible expansion to more equations with more variables.

   (Cramer's method:
   The system of equations can be written in matrix form as Ax = B, where:
   A - the coefficient matrix - the matrix made up by the coefficients of the variables on the left-hand side of the equation,
   x - the column matrix with variables
   B - the column matrix with the constants which are on the right side of the equations)

   Since the Cramer's method is a classic of university mathematics, this program can be used, for example, to calculate the number of loops by inserting counters in the appropriate places into the program. But undoubtedly the advantage of this program will be that the Cramer's method ceases to be a pure theory at this point and begins to exist as a living element of mathematics.

The programs are not short, but made of modules, that are copies of each other from the label T2 to label T10, systematically extended by only one element. The rest of the code is handling these labels. I am sure you will quickly notice the simplicity of the programs and be tempted to add your code into them.

These programs are easy to extend to square matrices with more elements (not just 10x10). The only limit here are... the patience of the user and his/her tolerance for the 'efficiency' of the computer.

Both programs are prepared with samples of data to immediate run them in the *Dev C++* environment. However, the lines for *Borland TurboC++* are commented out and clearly described.

Numerical data is declared as integer to make the code as clear as possible. In a general case, their floating-point class should be adopted. We should be aware that the determinant must be declared of the 'double' class to avoid unpleasant surprises of erroneous calculations (going beyond the range of the number type - then the compiler does not show any error!).

Results of running programs:

1. Calculation of a determinant of a quadratic matrix - max 10x10 of matrix dimensions

```
C:\Users\Leszek\Desktop\M_1.exe

Matrix:

   2   1  -1   3   1   0   2   1   0   2
   1   4  -3   2   5  -1  -1  -2   2   3
  -2   3   0   2   1   3   3   5   3  -1
   3  -1   5  -3   0   0   0   3   1   5
   1  -1   2   0   1  -2  -2   1  -1   0
  -1   3   0   2   4   1   1   3   2   2
   0   2  -1   3   2   1   1   4   0  -3
   5   2  -1   3   0  -3  -3  -1   1   2
   0   1   3   2   0  -1   1   1  -3  -2
   2  -2   1   3   1  -3   2   1   4   1

The determinant of the matrix is equal to -182974

Press any key to exit the program
```

2. Solving a system of 'n' linear equations with 'n' variables - up to 10 equations with 10 variables - Cramer method

```
C:\Users\Leszek\Desktop\M_2.exe

System of equations

2*x1 +1*x2 -1*x3 +3*x4 +1*x5 +0*x6 +2*x7 +1*x8 +0*x9 +2*x10  = 6
1*x1 +4*x2 -3*x3 +2*x4 +5*x5 -1*x6 -1*x7 -2*x8 +2*x9 +3*x10  = 15
-2*x1 +3*x2 +0*x3 +2*x4 +1*x5 +3*x6 +3*x7 +5*x8 +3*x9 -1*x10  = -17
3*x1 -1*x2 +5*x3 -3*x4 +0*x5 +0*x6 +0*x7 +3*x8 +1*x9 +5*x10  = 17
1*x1 -1*x2 +2*x3 +0*x4 +1*x5 -2*x6 -2*x7 +1*x8 -1*x9 +0*x10  = 17
-1*x1 +3*x2 +0*x3 +2*x4 +4*x5 +1*x6 +1*x7 +3*x8 +2*x9 +2*x10  = 8
0*x1 +2*x2 -1*x3 +3*x4 +2*x5 +1*x6 +1*x7 +4*x8 +0*x9 -3*x10  = -2
5*x1 +2*x2 -1*x3 +3*x4 +0*x5 -3*x6 -3*x7 -1*x8 +1*x9 +2*x10  = 16
0*x1 +1*x2 +3*x3 +2*x4 +0*x5 -1*x6 +1*x7 +1*x8 -3*x9 -2*x10  = 12
2*x1 -2*x2 +1*x3 +3*x4 +1*x5 -3*x6 +2*x7 +1*x8 +4*x9 +1*x10  = -1

The solution to this system of equations is:

        x1= 1.000       x2=-0.000       x3= 2.000       x4= 1.000
        x5= 3.000       x6=-1.000       x7=-2.000       x8=-0.000
        x9=-3.000       x10= 2.000


Press any key to exit the program
```

To verify results, please use RESHISH web page: https://matrix.reshish.com/determinant.php
Example: For point 1. (Calculation of a determinant of a quadratic matrix - max 10x10 of matrix dimensions)
Determinant Δ = -182974

Sign		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
+	1	2	1	-1	3	1	0	2	1	0	2
	2	0	7/2	-5/2	1/2	9/2	-1	-2	-5/2	2	2
	3	0	0	13/7	31/7	-22/7	29/7	51/7	62/7	5/7	-9/7
	4	0	0	0	-239/13	126/13	-146/13	-298/13	-296/13	8/13	87/13
	5	0	0	0	0	567/239	-657/239	-863/239	-376/239	-203/239	-206/239
	6	0	0	0	0	0	92/63	1186/567	158/567	4/81	1492/567
	7	0	0	0	0	0	0	-247/207	379/207	-187/207	-886/207
	8	0	0	0	0	0	0	0	-1000/247	281/494	2326/247
	9	0	0	0	0	0	0	0	0	-1683/250	-1243/125
	10	0	0	0	0	0	0	0	0	0	-8317/1224

$\Delta = 2 \times 7/2 \times 13/7 \times (-239/13) \times 567/239 \times 92/63 \times (-247/207) \times (-1000/247) \times (-1683/250) \times (-8317/1224) = -182974$

Hide solution Recalculate

Result:

Δ = -182974