

Домашнє завдання №23

Написати на C++ програму для розв'язання системи лінійних алгебраїчних рівнянь(СЛАР) з використанням бібліотеки **uBLAS**(з набору бібліотек **Boost**).

* *ручний розв'язок СЛАР не є обов'язковою умовою виконання завдання*

* **коментар:** далі наводиться приклад повністю виконаного завдання; також наведено приклад ручного розв'язання СЛАР для перевірки результатів виконання програми; для компіляції і запуску

* **рекомендується використати** https://www.onlinegdb.com/online_c++_compiler ; для компіляції на ПК слід скачати з https://www.boost.org/users/history/version_1_72_0.html та розархівувати набір бібліотек Boost, після чого налаштувати Visual Studio як показано в цій відео-демонстрації <https://www.youtube.com/watch?v=SWCCFGX6c0g>.

Вибір варіанту

$$(N_{\text{ж}} + N_{\text{г}} + 1) \% 20 + 1$$

де: $N_{\text{ж}}$ – порядковий номер студента в групі, а $N_{\text{г}}$ – номер групи(1,2,3,4,5,6,7,8 або 9)

Варіанти завдань

* **коментар:** для застосування наведеного далі прикладу коду x, y, z потрібно іменувати як x1, x2 та x3 відповідно

Варіант	СЛАР	Варіант	СЛАР
1	$\begin{cases} 5x - y + z = 0 \\ 3x + y + 2z = -1 \\ x + y + z = 0 \end{cases}$	2	$\begin{cases} 3x - 5y + z = 1 \\ 2x + y + 2z = 5 \\ x - y + z = 3 \end{cases}$
3	$\begin{cases} 5x + 7y + 4z = 3 \\ 3x - 5y + 2z = 3 \\ 2x + y + z = 3 \end{cases}$	4	$\begin{cases} 7x + y - z = 2 \\ 5x + 2y + z = 10 \\ 2x + 3y - 5z = 2 \end{cases}$
5	$\begin{cases} 3x + y + 2z = 6 \\ 2x - y + 7z = 7 \\ x + y - 3z = -2 \end{cases}$	6	$\begin{cases} 2x - 3y + z = 10 \\ x + y + 4z = 6 \\ x - y - 2z = 2 \end{cases}$

7	$\begin{cases} 3x + 2y - 2z = 1 \\ x + y + 3z = 6 \\ x - 2y - 7z = 1 \end{cases}$	8	$\begin{cases} 7x + y - z = 2 \\ 3x + 2y - 5z = 1 \\ 3x - y + 5z = 2 \end{cases}$
9	$\begin{cases} 5x + 3y - 4z = 8 \\ x - 3y + 5z = 6 \\ 2x - y + z = 4 \end{cases}$	10	$\begin{cases} 2x - 5y - z = 8 \\ x - y + 3z = 3 \\ 3x - 4y - 2z = 7 \end{cases}$
11	$\begin{cases} 2x + y - 3z = -1 \\ 3x + y - z = 1 \\ 5x + 2y + 4z = 0 \end{cases}$	12	$\begin{cases} 3x + y + 2z = 2 \\ 2x + 5y + 4z = 5 \\ 5x - y + 2z = 2 \end{cases}$
13	$\begin{cases} 2x - y - z = 5 \\ 3x - 5y - z = 5 \\ x + y + z = 7 \end{cases}$	14	$\begin{cases} 5x - 3y - 5z = 8 \\ 3x + 7y - 3z = 10 \\ x + 2y + 5z = 5 \end{cases}$
15	$\begin{cases} 5x + 5y + 2z = 1 \\ x + 2y + z = 4 \\ 3x + 7y - z = 2 \end{cases}$	16	$\begin{cases} 2x + y + z = 8 \\ 3x + 2y - 5z = 4 \\ 2x + 3y - z = 0 \end{cases}$
17	$\begin{cases} x - 3y - z = 0 \\ 2x - 5y + z = 5 \\ 3x - 7y - 5z = 2 \end{cases}$	18	$\begin{cases} 7x + y - z = 3 \\ 2x + 5y - z = 2 \\ 5x - y - z = 1 \end{cases}$
19	$\begin{cases} 5x + y + z = 6 \\ 2x - 3y - 2z = 2 \\ 3x + 2y + 2z = 5 \end{cases}$	20	$\begin{cases} 3x - 10y + 2z = 5 \\ 5x - y - 9z = 24 \\ 2x + 3y + 6z = 13 \end{cases}$

Приклад коду

Наведений зразок коду реалізовує розв’язання СЛАР з 4-х рівнянь для 4 невідомих.

* **коментар:** для заданих варіантів завдань, окрім визначення COEFFICIENTS та CONSTANT_TERMS, потрібно встановити для EQUATIONS_COUNT значення 3

СЛАР для прикладу		Макровизначення	
$\begin{cases} x_1 + 5x_2 + 3x_3 - 4x_4 = 20 \\ 3x_1 + x_2 - 2x_3 = 9 \\ 5x_1 - 7x_2 + 10x_4 = -9 \\ 3x_2 - 5x_3 = 1 \end{cases}$	\Rightarrow	$\begin{pmatrix} 1 & 5 & 3 & -4 \\ 3 & 1 & -2 & 0 \\ 5 & -7 & 0 & 10 \\ 0 & 3 & -5 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 20 \\ 9 \\ -9 \\ 1 \end{pmatrix}$	
		$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$	#define EQUATIONS_COUNT 4
		$\begin{pmatrix} 1 & 5 & 3 & -4 \\ 3 & 1 & -2 & 0 \\ 5 & -7 & 0 & 10 \\ 0 & 3 & -5 & 0 \end{pmatrix}$	#define COEFFICIENTS \ 1.0, 5.0, 3.0, -4.0, \ 3.0, 1.0, -2.0, 0.0, \ 5.0, -7.0, 0.0, 10.0, \ 0.0, 3.0, -5.0, 0.0
		$\begin{pmatrix} 20 \\ 9 \\ -9 \\ 1 \end{pmatrix}$	#define CONSTANT_TERMS \ 20.0, \ 9.0, \ -9.0, \ 1.0

Лістинг

```
//#include <vector> // not used !
#include <iostream>
#include <iomanip>

#include <boost/numeric/ublas/lu.hpp>
#include <boost/numeric/ublas/matrix.hpp>
#include <boost/numeric/ublas/vector.hpp>
#include <boost/numeric/ublas/io.hpp>

#define EQUATIONS_COUNT 4

#define COEFFICIENTS \
1.0, 5.0, 3.0, -4.0, \
3.0, 1.0, -2.0, 0.0, \
5.0, -7.0, 0.0, 10.0, \
0.0, 3.0, -5.0, 0.0

#define CONSTANT_TERMS \
20.0, \
9.0, \
-9.0, \
1.0

//#define DEBUG // uncomment to debug

using namespace boost::numeric::ublas;

#define DECL_VALUES_ARRAY(NAME, SIZE, ...)\
double arr_##NAME[SIZE] = { __VA_ARGS__ };\
std::vector<double> NAME;\
NAME.assign(arr_##NAME, arr_##NAME + SIZE);\
typedef matrix<double, row_major, std::vector<double>> Matrix;
```

```
typedef vector<double, std::vector<double>> Vector;

void printMatrix(const std::string & text, Matrix & printingMatrix){
    std::cout << text << "[" << printingMatrix.size1() << "]" << "[" <<
printingMatrix.size2() << "]" << std::endl;
    for(unsigned int iIndex = 0; iIndex < printingMatrix.size1(); ++iIndex){
        std::cout << "(" << " ";
        for(unsigned int jIndex = 0; jIndex < printingMatrix.size2(); ++jIndex){
            std::cout<<printingMatrix(iIndex, jIndex)<<" ";
        }
        std::cout << ")," << std::endl;
    }

    std::cout << ")" << std::endl << std::endl;
}

void printEquationsSystem(const std::string & text, Matrix & printingMatrix, Vector&
printingVector){
    std::cout << text << "{" << std::endl;
    for (unsigned int iIndex = 0; iIndex < printingMatrix.size1(); ++iIndex){
        unsigned char printedPrev = 0;
        for (unsigned int jIndex = 0; jIndex < printingMatrix.size2(); ++jIndex){
            double elementValue = printingMatrix(iIndex, jIndex);
            if (jIndex && printedPrev){
                if (elementValue > 0){
                    std::cout << " + ";
                }
                else if (elementValue < 0){
                    std::cout << " - ";
                }
                //else{} // nothing
            }
            if (elementValue > 0 || (!printedPrev && elementValue)){
                if (elementValue != -1. && elementValue != 1.) {
                    std::cout << printingMatrix(iIndex, jIndex) << "*";
                }
                std::cout << "X" << jIndex + 1; // << " ";
                printedPrev = ~0;
            }
            else if (printingMatrix(iIndex, jIndex) < 0){
                if (elementValue != -1.) {
                    std::cout << -printingMatrix(iIndex, jIndex) << "*";
                }
                std::cout << "X" << jIndex + 1; // << " ";
                printedPrev = ~0;
            }
            //else{} // nothing
        }
        std::cout << " = " << printingVector(iIndex) << ";" << std::endl;
    }

    std::cout << "}" << std::endl << std::endl;
}

void printSolutionSet(const std::string & text, Vector & printingVector){
    std::cout << text << "{" << std::endl;
    for (unsigned int index = 0; index < printingVector.size(); ++index){
        unsigned char printedPrev = 0;
        std::cout << "X" << index + 1 << " = " << printingVector(index) << ";" <<
std::endl;
    }

    std::cout << "}" << std::endl << std::endl;
}
```

```

Matrix invertMatrix(const Matrix & matrixA){
    Matrix copyMatrixA(matrixA);
    permutation_matrix<Matrix::size_type> permutationMatrixA(copyMatrixA.size1());
    size_t res = lu_factorize(copyMatrixA, permutationMatrixA);
    if(res != 0){
        throw std::logic_error("lu_factorize error");
    }

    Matrix inverseMatrixA(identity_matrix<double>(copyMatrixA.size1()));
    lu_substitute(copyMatrixA, permutationMatrixA, inverseMatrixA);

    return inverseMatrixA;
}

int main(int argc, char** argv){
    try{
        std::cout << std::fixed;

        // A
        DECL_VALUES_ARRAY(matrixA_values, EQUATIONS_COUNT * EQUATIONS_COUNT,
COEFFICIENTS);
        Matrix matrixA(EQUATIONS_COUNT, EQUATIONS_COUNT, matrixA_values);

        // b
        DECL_VALUES_ARRAY(vectorB_values, EQUATIONS_COUNT, CONSTANT_TERMS);
        Vector vectorB(EQUATIONS_COUNT, vectorB_values);

        // A^-1
        auto invertedMatrixA = invertMatrix(matrixA);

        // x = A^-1 * b
        Vector vectorX = prod(invertedMatrixA, vectorB);

#ifdef DEBUG
        printMatrix("matrixA: ", matrixA);
        std::cout << "vectorB: " << vectorB << std::endl << std::endl;
        printMatrix("invertedMatrixA: ", invertedMatrixA);
        std::cout << "vectorX: " << vectorX << std::endl << std::endl;
#endif

        printEquationsSystem("System of equations: ", matrixA, vectorB);
        printSolutionSet("Solving system of equations: ", vectorX);
    }
    catch(std::exception& ex){
        std::cout << ex.what() << std::endl;
    }

#ifdef __linux__
    std::cout << "Press any key to continue . . . " << std::endl;
    (void)getchar();
#elif defined(_WIN32)
    system("pause");
#else
#endif

    return 0;
}

```

Оскільки він відмінний від нуля, то задана **система рівнянь** сумісна і має єдиний розв'язок. На наступному кроці, приступимо до знаходження оберненої матриці. Для цього скористаємось другим з перерахованих вище методів, а саме методом алгебраїчних доповнень, та, слідуючи його алгоритму, для матриці коефіцієнтів **знайдемо транспоновану матрицю**, після чого для елементів отриманої матриці обчислимо алгебраїчні доповнення:

$$A^T = \begin{pmatrix} 1 & 3 & 5 & 0 \\ 5 & 1 & -7 & 3 \\ 3 & -2 & 0 & -5 \\ -4 & 0 & 10 & 0 \end{pmatrix}$$

$$A_{11} = (-1)^2 \cdot \begin{vmatrix} 1 & -7 & 3 \\ -2 & 0 & -5 \\ 0 & 10 & 0 \end{vmatrix} = (1 \cdot 0 \cdot 0) + ((-7) \cdot (-5) \cdot 0) + (3 \cdot (-2) \cdot 10) - (3 \cdot 0 \cdot 0) - \\ - ((-7) \cdot (-2) \cdot 0) - (1 \cdot (-5) \cdot 10) = -10$$

$$A_{12} = (-1)^3 \cdot \begin{vmatrix} 5 & -7 & 3 \\ 3 & 0 & -5 \\ -4 & 10 & 0 \end{vmatrix} = -((5 \cdot 0 \cdot 0) + ((-7) \cdot (-5) \cdot (-4)) + (3 \cdot 3 \cdot 10) - (3 \cdot 0 \cdot (-4)) - \\ - ((-7) \cdot 3 \cdot 0) - (5 \cdot (-5) \cdot 10)) = -200$$

$$A_{13} = (-1)^4 \cdot \begin{vmatrix} 5 & 1 & 3 \\ 3 & -2 & -5 \\ -4 & 0 & 0 \end{vmatrix} = (5 \cdot (-2) \cdot 0) + (1 \cdot (-5) \cdot (-4)) + (3 \cdot 3 \cdot 0) - (3 \cdot (-2) \cdot (-4)) - \\ - (1 \cdot 3 \cdot 0) - (5 \cdot (-5) \cdot 0) = -4$$

$$A_{14} = (-1)^5 \cdot \begin{vmatrix} 5 & 1 & -7 \\ 3 & -2 & 0 \\ -4 & 0 & 10 \end{vmatrix} = -((5 \cdot (-2) \cdot 10) + (1 \cdot 0 \cdot (-4)) + ((-7) \cdot 3 \cdot 0) - ((-7) \cdot (-2) \cdot (-4)) - \\ - (1 \cdot 3 \cdot 10) - (5 \cdot 0 \cdot 0)) = 74$$

$$A_{21} = (-1)^3 \cdot \begin{vmatrix} 3 & 5 & 0 \\ -2 & 0 & -5 \\ 0 & 10 & 0 \end{vmatrix} = -((3 \cdot 0 \cdot 0) + (5 \cdot (-5) \cdot 0) + (0 \cdot (-2) \cdot 10) - (0 \cdot 0 \cdot 0) - \\ - (5 \cdot (-2) \cdot 0) - (3 \cdot (-5) \cdot 10)) = -150$$

$$A_{22} = (-1)^4 \cdot \begin{vmatrix} 1 & 5 & 0 \\ 3 & 0 & -5 \\ -4 & 10 & 0 \end{vmatrix} = (1 \cdot 0 \cdot 0) + (5 \cdot (-5) \cdot (-4)) + (0 \cdot 3 \cdot 10) - (0 \cdot 0 \cdot (-4)) - \\ - (5 \cdot 3 \cdot 0) - (1 \cdot (-5) \cdot 10) = 150$$

$$A_{23} = (-1)^5 \cdot \begin{vmatrix} 1 & 3 & 0 \\ 3 & -2 & -5 \\ -4 & 0 & 0 \end{vmatrix} = -((1 \cdot (-2) \cdot 0) + (3 \cdot (-5) \cdot (-4)) + (0 \cdot 3 \cdot 0) - (0 \cdot (-2) \cdot (-4)) - \\ - (3 \cdot 3 \cdot 0) - (1 \cdot (-5) \cdot 0)) = -60$$

$$A_{24} = (-1)^6 \cdot \begin{vmatrix} 1 & 3 & 5 \\ 3 & -2 & 0 \\ -4 & 0 & 10 \end{vmatrix} = (1 \cdot (-2) \cdot 10) + (3 \cdot 0 \cdot (-4)) + (5 \cdot 3 \cdot 0) - (5 \cdot (-2) \cdot (-4)) - \\ - (3 \cdot 3 \cdot 10) - (1 \cdot 0 \cdot 0) = -150$$

$$A_{31} = (-1)^4 \cdot \begin{vmatrix} 3 & 5 & 0 \\ 1 & -7 & 3 \\ 0 & 10 & 0 \end{vmatrix} = (3 \cdot (-7) \cdot 0) + (5 \cdot 3 \cdot 0) + (0 \cdot 1 \cdot 10) - (0 \cdot (-7) \cdot 0) - \\ - (5 \cdot 1 \cdot 0) - (3 \cdot 3 \cdot 10) = -90$$

$$A_{32} = (-1)^5 \cdot \begin{vmatrix} 1 & 5 & 0 \\ 5 & -7 & 3 \\ -4 & 10 & 0 \end{vmatrix} = -((1 \cdot (-7) \cdot 0) + (5 \cdot 3 \cdot (-4)) + (0 \cdot 5 \cdot 10) - (0 \cdot (-7) \cdot (-4)) - \\ - (5 \cdot 5 \cdot 0) - (1 \cdot 3 \cdot 10)) = 90$$

$$A_{33} = (-1)^6 \cdot \begin{vmatrix} 1 & 3 & 0 \\ 5 & 1 & 3 \\ -4 & 0 & 0 \end{vmatrix} = (1 \cdot 1 \cdot 0) + (3 \cdot 3 \cdot (-4)) + (0 \cdot 5 \cdot 0) - (0 \cdot 1 \cdot (-4)) - \\ - (3 \cdot 5 \cdot 0) - (1 \cdot 3 \cdot 0) = -36$$

$$A_{34} = (-1)^7 \cdot \begin{vmatrix} 1 & 3 & 5 \\ 5 & 1 & -7 \\ -4 & 0 & 10 \end{vmatrix} = -((1 \cdot 1 \cdot 10) + (3 \cdot (-7) \cdot (-4)) + (5 \cdot 5 \cdot 0) - (5 \cdot 1 \cdot (-4)) - \\ - (3 \cdot 5 \cdot 10) - (1 \cdot (-7) \cdot 0)) = 36$$

$$A_{41} = (-1)^5 \cdot \begin{vmatrix} 3 & 5 & 0 \\ 1 & -7 & 3 \\ -2 & 0 & -5 \end{vmatrix} = -((3 \cdot (-7) \cdot (-5)) + (5 \cdot 3 \cdot (-2)) + (0 \cdot 1 \cdot 0) - (0 \cdot (-7) \cdot (-2)) - \\ - (5 \cdot 1 \cdot (-5)) - (3 \cdot 3 \cdot 0)) = -100$$

$$A_{42} = (-1)^6 \cdot \begin{vmatrix} 1 & 5 & 0 \\ 5 & -7 & 3 \\ 3 & 0 & -5 \end{vmatrix} = (1 \cdot (-7) \cdot (-5)) + (5 \cdot 3 \cdot 3) + (0 \cdot 5 \cdot 0) - (0 \cdot (-7) \cdot 3) - \\ - (5 \cdot 5 \cdot (-5)) - (1 \cdot 3 \cdot 0) = 205$$

$$A_{43} = (-1)^7 \cdot \begin{vmatrix} 1 & 3 & 0 \\ 5 & 1 & 3 \\ 3 & -2 & -5 \end{vmatrix} = -((1 \cdot 1 \cdot (-5)) + (3 \cdot 3 \cdot 3) + (0 \cdot 5 \cdot (-2)) - (0 \cdot 1 \cdot 3) - \\ - (3 \cdot 5 \cdot (-5)) - (1 \cdot 3 \cdot (-2))) = -103$$

$$A_{44} = (-1)^8 \cdot \begin{vmatrix} 1 & 3 & 5 \\ 5 & 1 & -7 \\ 3 & -2 & 0 \end{vmatrix} = (1 \cdot 1 \cdot 0) + (3 \cdot (-7) \cdot 3) + (5 \cdot 5 \cdot (-2)) - (5 \cdot 1 \cdot 3) - \\ - (3 \cdot 5 \cdot 0) - (1 \cdot (-7) \cdot (-2)) = -142$$

Після того, як алгебраїчні доповнення відомі, обернену матрицю знаходимо за наступною формулою:

$$A^{-1} = \frac{1}{\Delta} \cdot \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} = \frac{1}{-630} \cdot \begin{pmatrix} -10 & -200 & -4 & 74 \\ -150 & 150 & -60 & -150 \\ -90 & 90 & -36 & 36 \\ -100 & 205 & -103 & -142 \end{pmatrix} = \begin{pmatrix} 0.015 & 0.317 & 0.006 & -0.117 \\ 0.238 & -0.238 & 0.095 & 0.238 \\ 0.143 & -0.143 & 0.057 & -0.057 \\ 0.159 & -0.325 & 0.163 & 0.225 \end{pmatrix}$$

Далі, скориставшись формулою (3) знаходимо шуканий розв'язок заданої системи:

$$x = A^{-1} \cdot b = \begin{pmatrix} 0.015 & 0.317 & 0.006 & -0.117 \\ 0.238 & -0.238 & 0.095 & 0.238 \\ 0.143 & -0.143 & 0.057 & -0.057 \\ 0.159 & -0.325 & 0.163 & 0.225 \end{pmatrix} \cdot \begin{pmatrix} 20 \\ 9 \\ -9 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 1 \\ -1 \end{pmatrix}$$

Тобто, $x_1=3$, $x_2=2$, $x_3=1$, $x_4=-1$.