НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ “КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп’ютерних систем

Лабораторна робота №3

з дисципліни “Алгоритми та методи обчислень”

Тема: “ **РОЗВ’ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ АЛГЕБРАЇЧНИХ РІВНЯНЬ** ”

Варіант 4

Виконала: студентка III курсу

КВ-72 Дорош Карина

Перевірив: Зорін Ю. М.

Київ 2019

Завдання

4 – виключення Гаусса-Жордана, метод ітерації Зейделя;

4	3	7	6	0	63
	13	49	20	15	353
	20	19	50	10	437
	17	7	17	17	225

Код програми:

main.cpp

```
#include "header.h"

int main()
{
	double initial_matrix[N][M] = { {3,  7, 6, 0,  63},
                                        {13,  49, 20,  15, 353},
                                        {20, 19,  50, 10, 437},
                                        {17, 7, 17, 17,  225} };
	double modified_matrix[N][M] = { {61, 1, -2, 17, 181},
                                        {13, 49, 20, 15, 353},
                                        {20, 19, 50, 10, 437},
                                        {31, 2, 1, 41, 238} };
	double Gequation_root[N] = { 0, 0, 0, 0 };
	double Sequation_root[N] = { 0, 0, 0, 0 };
	double error = 0, eps = 0.001;
	cout << "Gauss method:" << endl;
	print_matrix(*initial_matrix);
	if (gauss(initial_matrix, Gequation_root)) {
		cout << "SLAR was solved!\n\nRoots are ";
		for (auto i = 0; i < N; i++)
			cout << Gequation_root[i] << " ";
	}
	else {
		cout << "Error\n\n" << endl;
	}
	cout << "\n\nIteration Seidel method:" << endl;
	print_matrix(*modified_matrix);
	if (seidelMethod(modified_matrix, Sequation_root, error, eps)) {
		cout << "\nSLAR was solved!\n\nRoots are ";
		for (auto i = 0; i < N; i++)
			cout << Sequation_root[i] << " ";
		cout << "\nInfelicity in calculations: " << error << endl;
	}
	else {
		cout << "Error" << endl;
	}
	return 0;
}
```

holving.cpp

```cpp
#include "header.h"

void print_matrix(double* slar) {
	for (int i = 0; i < N; i++) {
		for (int j = 0; j < M; j++)
			cout << setw(4) << slar[i * M + j] << " ";
		cout << endl;
	}
	cout << endl;
}

bool gauss(double(*slar)[M], double* ans) {
	if ((!simplifyRow(slar, 0)) || (!simplifyRow(slar, 1)) || (!simplifyRow(slar,
2)) || (!simplifyRow(slar, 3))) {
		return false;
	}
	else{
		substr(slar, 0, 1);
		substr(slar, 0, 2);
		substr(slar, 0, 3);
		substr(slar, 1, 2);
		substr(slar, 1, 3);
		substr(slar, 2, 3);
		for (auto i = 0; i < N; ++i)
			ans[i] = slar[i][M - 1];
		return true;
	}
}
void substr(double(*slar)[M], int from, int what) {
	double coef = slar[from][what];
	for (auto i = what; i < M; ++i)
		slar[from][i] -= slar[what][i] * coef;
}
bool simplifyRow(double(*slar)[M], int row) {
	if (!slar[row][row]) {
		return false;
	}
	double firstElement = slar[row][row];
	for (auto i = row; i < M; ++i) {
		slar[row][i] /= !firstElement ? 1.0 : firstElement;
	}
	for (int line = row + 1; line < N; ++line) {
		auto firstElement = slar[line][row];
		for (int column = 0; column < M; ++column) {
			slar[line][column] -= firstElement * slar[row][column];
		}
	}
	//print_matrix(*slar);
	return true;
}
bool seidelMethod(double(*slar)[M], double* answer, double& error, double& eps) {
	double sum1 = 0, sum2 = 0, q = 0, max = 0, prev = 0;
	double temp_matrix[N][M];
	double temp = 0;
	double x_norm = 0;
```

```
	for (auto i = 0; i < N; ++i) {
		if (slar[i][i] == 0) return false;
		temp_matrix[i][i] = slar[i][M-1]/slar[i][i];
		temp = slar[i][i];
		sum2 = 0;
		for (auto j = 0; j < M-1; ++j) {
			if (j == i) continue;
			temp_matrix[i][j] = slar[i][j] / temp;
			sum2 += temp_matrix[i][j];
		}
		if ((sum2 > max) && (sum2 < 1)) {
			max = sum2;
			q = max;
		}
	}
	cout << "Q = " << q;
	do {
		for (auto i = 0; i < N; ++i) {
			sum1 = 0;
			for (auto j = 0; j < M - 1; ++j) {
				if (j == i) continue;
				sum1 += answer[j] * temp_matrix[i][j];
			}
			answer[i] = temp_matrix[i][i] - sum1;
		}
		x_norm = answer[N - 1] - prev;
		error = abs(x_norm);
		prev = answer[N - 1];
	} while (x_norm > (1 - q) / q * eps);

	return true;
}
```

header.h

```
#pragma once
#include <iostream>
#include <Windows.h>
#include <stdlib.h>
#include <iomanip>
#include <vector>
#include <algorithm>
#define N 4
#define M 5
#define maxIteration 200
using namespace std;

void print_matrix(double* matrix);
bool gauss(double(*slar)[M], double* ans);

void substr(double(*slar)[M], int from, int what);
bool simplifyRow(double(*slar)[M], int row);
bool seidelMethod(double(*slar)[M], double* answer, double& error, double& eps);
```

Приклад роботи програми