ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ

УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра Інформатики

**Звіт**

з лабораторної роботи № 3

на тему «рішення СЛАР»

Виконав Перевірив

ст.гр.ІТІНФ-20-1 ас.каф. Інформатики

Самченко С.О. Пономаренко Р.П.

Харків 2021

**Мета роботи:** Закріплення практичних навичок розв'язання систем лінійних алгебраїчних рівнянь (СЛАР) точними і наближеними методами.

Варіант 21(5): методи: Зейделя і головних елементів

**Метод Зейделя:**

#include <iostream>

#include <cmath>

using namespace std;

// проверка на сходимость

bool converge(double xk[3], double xkp[3], int n, double eps) {

double norm = 0;

for (int i = 0; i < n; i++)

norm += (xk[i] - xkp[i]) \* (xk[i] - xkp[i]);

return (sqrt(norm) < eps);

}

// формула вычисления і-го компонента (k + 1)-го приближения

double okr(double x, double eps) {

int i = 0;

double neweps = eps;

while (neweps < 1) {

i++;

neweps \*= 10;

}

int okr = pow(double(10), i);

x = int(x \* okr + 0.5) / double(okr);

return x;

}

// сравнение значения элемента главной диагонали и суммы остальных элементов в строке по модулю

bool diagonal(double a[3][3], int n) {

int i, j, k = 1;

double sum;

for (i = 0; i < n; i++) {

sum = 0;

for (j = 0; j < n; j++) sum += abs(a[i][j]);

sum -= abs(a[i][i]);

if (sum > abs(a[i][i])) {

k = 0;

cout << a[i][i] << " < " << sum << endl;

}

else {

cout << a[i][i] << " > " << sum << endl;

}

}

return (k == 1);

}

int main() {

setlocale(LC\_ALL, "");

double a[3][3] = { -5, -1, 3, 3, 6, -1, 1, -1, 5 }, b[3] = { 21, -4, 17 }, eps(0.0000001), x[3], p[3];

int n(3), m = 0;

int method;

cout << "eps = " << eps << endl;

cout << "матрица А: " << endl << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++)

cout << a[i][j] << " ";

cout << endl;

}

cout << endl << "столбец свободных членов: " << endl << endl;

for (int i = 0; i < n; i++) {

cout << b[i] << " ";

}

cout << endl << endl;

for (int i = 0; i < n; i++)

x[i] = 1;

cout << "Диагональное преобладание (по модулю): " << endl;

if (diagonal(a, n)) {

do {// итерационный процесс

for (int i = 0; i < n; i++)

p[i] = x[i];

for (int i = 0; i < n; i++) {

double var = 0;

for (int j = 0; j < n; j++)

if (j != i) var += (a[i][j] \* x[j]);

x[i] = (b[i] - var) / a[i][i];

}

m++;

} while (!converge(x, p, n, eps));

cout << "Решение системы:" << endl << endl;

for (int i = 0; i < n; i++) cout << "x" << i + 1 << " = " << okr(x[i], eps) << "" << endl;

cout << "Итераций: " << m << endl;

}

else {

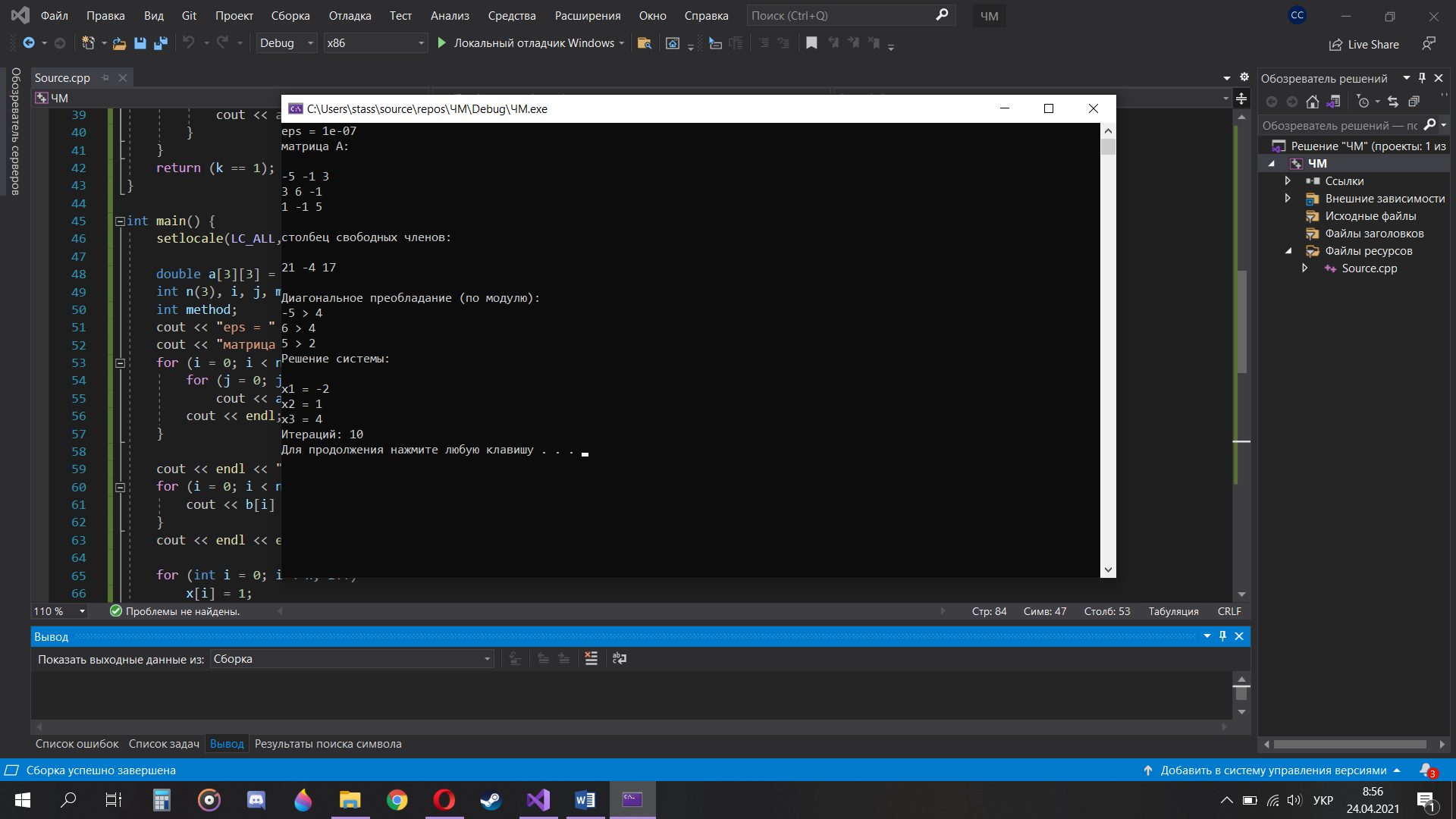
cout << "Не выполняется преобладание диагоналей" << endl;

}

system("pause");

return 0;

}



**Метод головних елементів:**

#include <stdlib.h>

#include <iostream>

#include <cmath>

#define N 3

#define N1 N+1

using namespace std;

float matrix[N][N1] = { {-5,-1,3,21}, {3,6,-1,-4}, {1,-1,5,17} };

void ShowMatrix(void) {

cout << "матрица А: " << endl << endl;

int i, j;

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++)

cout << matrix[i][j] << " ";

cout << endl;

}

cout << endl << "столбец свободных членов: " << endl << endl;

for (i = 0; i < N; i++) {

for (j = 0; j < N1 - 3; j++)

cout << matrix[i][j + 3] << " ";

}

cout << endl;

}

int main() {

setlocale(LC\_ALL, "");

float tmp, xx[N1];

short int i, j, k;

ShowMatrix();

// преобразование к ступенчастому виду

for (i = 0; i < N; i++) {

tmp = matrix[i][i];

for (j = N; j >= i; j--)

matrix[i][j] /= tmp;

for (j = i + 1; j < N; j++) {

tmp = matrix[j][i];

for (k = N; k >= i; k--)

matrix[j][k] -= tmp \* matrix[i][k];

}

}

// нахождение х-ов обратным ходом

xx[N - 1] = matrix[N - 1][N];

for (i = N - 2; i >= 0; i--) {

xx[i] = matrix[i][N];

for (j = i + 1; j < N; j++) xx[i] -= matrix[i][j] \* xx[j];

}

// вывод решения

printf("\nзначения аргуметов:\n");

for (i = 0; i < N; i++)

printf("x%d=%3.3f\n", i + 1, xx[i]);

system("pause");

return 0;

}

