

МИНЕСТЕРСТВО НАУКИ ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования

**"КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ"**

Факультет Математики и Компьютерных Наук
Направление Математики и Компьютерных Наук

Лабораторная работа №3
Вариант №17

Работу выполнил _____ Батурин Н.Ю.

Специальность 02.03.01 математика и компьютерные науки курс 2

Специализация _____

Преподаватель _____ Виноградова К.Н.

Краснодар
2023

Содержание

1	Задание №1	3
1.1	Условие:	3
1.2	Код:	3
1.3	Результат:	3
2	Задание №2	4
2.1	Условие:	4
2.2	Код:	4
2.3	Результат:	5
3	Задание №3	6
3.1	Условие:	6
3.2	Код:	6
3.3	Результат:	7
3.4	Проверка через мат пакеты:	8

1 Задание №1

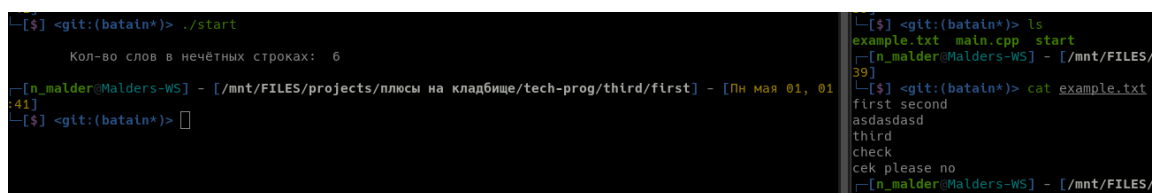
1.1 Условие:

Определить количество слов в нечётных строках текста.

1.2 Код:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main() {
    ifstream file("example.txt");
    if (file.is_open()) {
        string word;
        char symbol;
        int kol_str = 1, kol_words = 0;
        while (!file.eof()) {
            file >> word;
            if (kol_str % 2 != 0) kol_words++;
            file.get(symbol);
            if (symbol == '\n') kol_str++;
        }
        cout << "Кол-во слов в нечётных строках: " << kol_words << endl;
        file.close();
    }
    return 0;
}
```

1.3 Результат:



```
[n_malder@Malders-WS] - [/mnt/FILES/projects/плюсы на кладбище/tech-prog/third/first] - [Пн мая 01, 01:41]
--[$] <git:(batain*)> ./start
Кол-во слов в нечётных строках: 6
--[$] <git:(batain*)> []

--[$] <git:(batain*)> ls
example.txt main.cpp start
--[$] <git:(batain*)> cat example.txt
first second
asdasdasd
third
check
cek please no
--[$] <git:(batain*)>
```

2 Задание №2

2.1 Условие:

Даны 3 комплексных числа. Посчитать без использования библиотеки `complex` $x = \frac{a+b^3+c}{a-b^2-c}$

2.2 Код:

```
#include <iostream>
using namespace std;
struct Complex { float Im; float Re; };
Complex Plus(Complex first, Complex second) {
    Complex summa;
    summa.Im = first.Im + second.Im;
    summa.Re = first.Re + second.Re;
    return summa;
}
Complex Minus(Complex first, Complex second) {
    Complex difference;
    difference.Im = first.Im - second.Im;
    difference.Re = first.Re - second.Re;
    return difference;
}
Complex Multiply(Complex first, Complex second) {
    Complex piece;
    piece.Re = first.Re * second.Re - first.Im * second.Im;
    piece.Im = first.Re * second.Im + second.Re * first.Im;
    return piece;
}
Complex Share(Complex first, Complex second) {
    Complex division;
    division.Re = (first.Re * second.Re + first.Im * second.Im) / (pow(second.Re, 2) + pow(second.Im, 2));
    division.Im = (second.Re * first.Im - first.Re * second.Im) / (pow(second.Re, 2) + pow(second.Im, 2));
    return division;
}
int main() {
    Complex x, num, den, a, b, c, buffer;
    cout << "Последовательно введите действительную и мнимую часть для\на: ";
    cin >> a.Im >> a.Re;
    cout << "b: "; cin >> b.Im >> b.Re;
    cout << "c: "; cin >> c.Im >> c.Re;
    buffer = Multiply(b, b);
    num = Plus(Plus(a, c), Multiply(buffer, b));
    den = Minus(Minus(a, buffer), c);
    x = Share(num, den);
    cout << "\nX.Re: " << x.Re << "\nX.Im: " << x.Im << endl;
    return 0;
}
```

2.3 Результат:

```
Последовательно введите действительную и мнимую часть для  
a: 40 68  
b: 57 23  
c: 69 9  
  
X.Re: -56.9386  
X.Im: -24.066
```

3 Задание №3

3.1 Условие:

Для заданных матриц комплексных чисел $A(n \times n)$ и $B(n \times n)$ найти $C = (A^2 + B^2)^T$.

Вычислить C^{-1}

3.2 Код:

```
#include <iostream>
#include <fstream>
#include <cmath>
#include <complex>
#include <string>
using namespace std;
complex<int> Determ(complex<int>** matr, int n) {
    complex<int> det;
    if (n == 1) det = matr[n - 1][n - 1];
    if (n == 2)
        det = matr[0][0] * matr[1][1] - matr[0][1] * matr[1][0];
    if (n == 3) {
        det = matr[0][0] * matr[1][1] * matr[2][2] + matr[0][1] * matr[1][2] * matr[2][0] + matr[1][0] * matr[2][1] * matr[0][2] -
            det -= matr[2][0] * matr[1][1] * matr[0][2] - matr[1][0] * matr[0][1] * matr[2][2] - matr[2][1] * matr[1][2] * matr[0][0];
    }
    return det;
}

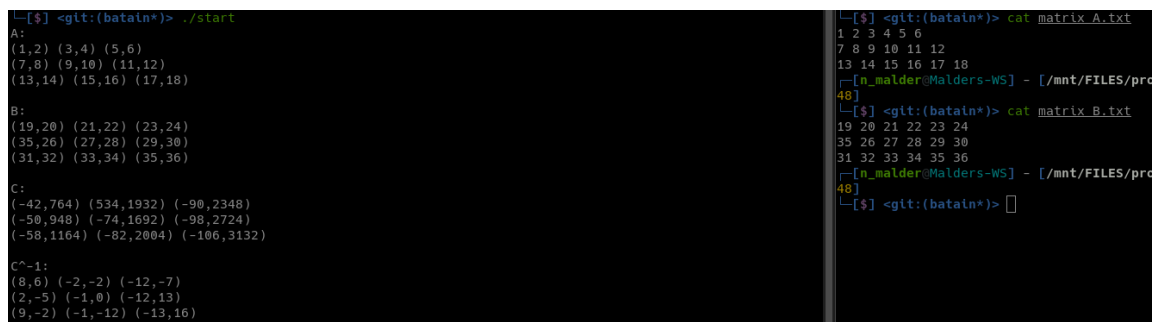
int main() {
    ifstream size_matrix("matrix_A.txt"), matr_A("matrix_A.txt"), matr_B("matrix_B.txt");
    if (size_matrix.is_open() && matr_A.is_open() && matr_B.is_open()) {
        int n = 0, buf;
        string number;
        char symbol;
        while (symbol != '\n') {
            size_matrix >> number; n++;
            size_matrix.get(symbol);
        }
        n /= 2;
        size_matrix.close();
        complex<int>** A = new complex<int>*[n], ** B = new complex<int>*[n], ** C_T = new complex<int>*[n], ** C = new complex<int>*[n];
        for (int i = 0; i < n; i++) {
            A[i] = new complex<int>[n];
            B[i] = new complex<int>[n];
            C_T[i] = new complex<int>[n];
            C[i] = new complex<int>[n];
            for (int j = 0; j < n; j++) {
                matr_A >> buf; A[i][j].real(buf);
                matr_A >> buf; A[i][j].imag(buf);
                matr_B >> buf; B[i][j].real(buf);
                matr_B >> buf; B[i][j].imag(buf);
            }
        }
        matr_A.close();
        matr_B.close();
        cout << "A:\n";
```

```

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) cout << A[i][j] << " ";
        cout << endl;
    }
    cout << "\nB:\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) cout << B[i][j] << " ";
        cout << endl;
    }
    cout << "\nC:\n";
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            C_T[j][i] = A[i][j] * A[i][j] + B[i][j] * B[i][j];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) cout << C_T[i][j] << " ";
        cout << endl;
    }
    cout << "\nC^-1:\n";
    complex<int> det = Determ(C_T, n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            C[i][j] = C_T[j][i] / det;
            cout << C[i][j] << " ";
        }
        cout << endl;
    }
    for (int i = 0; i < n; i++) {
        delete[]A[i]; delete[]B[i];
        delete[]C_T[i]; delete[]C[i];
    }
    delete[]A; delete[]B;
    delete[]C_T; delete[]C;
}
return 0;
}

```

3.3 Результат:



```

[~]$ <git:(batain*)> ./start
A:
(1,2) (3,4) (5,6)
(7,8) (9,10) (11,12)
(13,14) (15,16) (17,18)
B:
(19,20) (21,22) (23,24)
(35,26) (27,28) (29,30)
(31,32) (33,34) (35,36)
C:
(-42,764) (534,1932) (-90,2348)
(-50,948) (-74,1692) (-98,2724)
(-58,1164) (-82,2004) (-106,3132)
C^-1:
(8,6) (-2,-2) (-12,-7)
(2,-5) (-1,0) (-12,13)
(9,-2) (-1,-12) (-13,16)
[~]$ <git:(batain*)> cat matrix_A.txt
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
[~]$ <git:(batain*)> cat matrix_B.txt
19 20 21 22 23 24
35 26 27 28 29 30
31 32 33 34 35 36
[~]$ <git:(batain*)>

```

3.4 Проверка через мат пакеты:

```
Octave was configured for "x86_64-pc-linux-gnu".

Additional information about Octave is available at https://www.octave.org.

Please contribute if you find this software useful.
For more information, visit https://www.octave.org/get-involved.html

Read https://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

octave:1> A = [1+2*i 3+4*i 5+6*i; 7+8*i 9+10*i 11+12*i; 13+14*i 15+16*i 17+18*i]
A =

    1 + 2i    3 + 4i    5 + 6i
    7 + 8i    9 + 10i   11 + 12i
   13 + 14i   15 + 16i   17 + 18i

octave:2> B = [19+20*i 21+22*i 23+24*i; 25+26*i 27+28*i 29+30*i; 31+32*i 33+34*i 35+36*i]
B =

   19 + 20i   21 + 22i   23 + 24i
   25 + 26i   27 + 28i   29 + 30i
   31 + 32i   33 + 34i   35 + 36i

octave:3> C = (A^2 + B^2)'
C =

  -42 - 764i    534 + 1932i    -90 + 238i
  -50 + 948i    -74 + 1692i    -98 + 2724i
  -58 + 1164i   -82 + 2004i   -106 + 3132i

octave:4> C^-1
ans =

    8 + 6i    -2 - 2i    -12 - 7i
    2 - 5i    -1 + 0i    -12 + 13i
    9 - 2i    -1 - 12i   -13 + 16i
```