

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:	9
3.4	Проверка через мат пакеты:	9

1 Задание №1

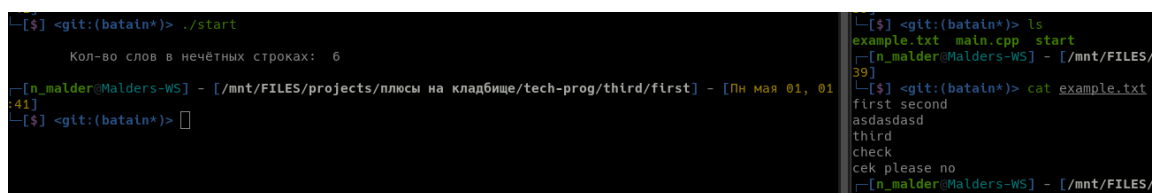
1.1 Условие:

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*)> []

[n_malder@Malders-WS] - [/mnt/FILES/projects/плюсы на кладбище/tech-prog/third/first] - [Пн мая 01, 01:41]
--[$] <git:(batain*)> ls
example.txt main.cpp start
--[$] <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 <complex>
#include <cmath>
using namespace std;
int SLAU(complex <float>** matrica_a, int n, complex <float>* massiv_b, complex <float>* x){
    int i, j, k, r;
    complex <float> c, M, s;
    float max;
    complex <float>** a, * b;
    a = new complex <float>*[n];
    for (i = 0; i < n; i++)
        a[i] = new complex <float>[n];
    b = new complex <float>[n];
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            a[i][j] = matrica_a[i][j];
    for (i = 0; i < n; i++)
        b[i] = massiv_b[i];
    for (k = 0; k < n; k++){
        max = abs(a[k][k]);
        r = k;
        for (i = k + 1; i < n; i++){
            if (abs(a[i][k]) > max){
                max = abs(a[i][k]);
                r = i;
            }
        }
        for (j = 0; j < n; j++){
            c = a[k][j];
            a[k][j] = a[r][j];
            a[r][j] = c;
        }
        c = b[k];
        b[k] = b[r];
        b[r] = c;
        for (i = k + 1; i < n; i++){
            for (M = a[i][k] / a[k][k], j = k; j < n; j++){
                a[i][j] -= M * a[k][j];
                b[i] -= M * b[k];
            }
        }
    }
    if (abs(a[n - 1][n - 1]) == 0)
        if (abs(b[n - 1]) == 0)
            return -1;
        else return -2;
```

```

else{
    for (i = n - 1; i >= 0; i--){
        for (s = 0, j = i + 1; j < n; j++){
            s += a[i][j] * x[j];
            x[i] = (b[i] - s) / a[i][i];
        }
        return 0;
    }
    for (i = 0; i < n; i++){
        delete[] a[i];
    }
    delete[] a;
    delete[] b;
}

int INVERSE(complex <float>** a, int n, complex <float>** y){
    int i, j, res;
    complex <float>* b, * x;
    b = new complex <float>[n];
    x = new complex <float>[n];
    for (i = 0; i < n; i++){
        for (j = 0; j < n; j++){
            if (j == i)
                b[j] = 1;
            else b[j] = 0;
        }
        res = SLAU(a, n, b, x);
        if (res != 0)
            break;
        else
            for (j = 0; j < n; j++){
                y[j][i] = x[j];
            }
        delete[] x;
        delete[] b;
        if (res != 0)
            return -1;
        else
            return 0;
    }
}

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 size = 0, buf;
        string number;
        char symbol;
        while (symbol != '\n') {
            size_matrix >> number; size++;
            size_matrix.get(symbol);
        }
        size /= 2;
        size_matrix.close();

        complex<float>** A = new complex<float> *[size], ** B = new complex<float> *[size], ** prom_C = new complex<float>
        for (int i = 0; i < size; i++) {
            A[i] = new complex<float>[size];
            B[i] = new complex<float>[size];
            prom_C[i] = new complex<float>[size];
            reverse_C[i] = new complex<float>[size];
            C[i] = new complex<float>[size];
            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 < size; i++) {
    for (int j = 0; j < size; j++) cout << A[i][j] << " ";
    cout << endl;
}
cout << "\nB:\n";
for (int i = 0; i < size; i++) {
    for (int j = 0; j < size; j++) cout << B[i][j] << " ";
    cout << endl;
}
cout << "\nC:\n";
for (int i = 0; i < size; i++)
    for (int j = 0; j < size; j++) {
        A[i][j] *= A[i][j];
        B[i][j] *= B[i][j];
        prom_C[i][j] = A[i][j] + B[i][j];
    }
for (int i = 0; i < size; i++)
    for (int j = 0; j < size; j++)
        C[j][i] = prom_C[i][j];
cout << "\nC:\n";
for (int i = 0; i < size; i++) {
    for (int j = 0; j < size; j++) cout << C[i][j] << " ";
    cout << endl;
}

if (!INVERSE(C, size, reverse_C)) {
    cout << endl << "Обратная матрица C:" << endl;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
            cout << reverse_C[i][j] << " ";
        cout << endl;
    }
}

for (int i = 0; i < size; i++) {
    delete[] A[i];
    delete[] B[i];
    delete[] C[i];
    delete[] prom_C[i];
    delete[] reverse_C[i];
}
delete[] A;
delete[] B;
delete[] C;
delete[] reverse_C;
delete[] prom_C;
}
return 0;
}

```


3.3 Результат:

```
n_malder@WS-Maldera:mnt/FILES/projects/нмсы на кладбище/tech-prog/third 10x53
n_malder@WS-Maldera: /mnt/FILES/projects/нмсы на кладбище/tech-prog/third/third - [Вт мая 28, 10:29]
--($)-->git:(batain+)> g++ main.cpp -lm -o start && ./start
A:
(6,2) (1,12) (7,6)
(14,3) (12,4) (7,3)
(7,8) (7,2) (3,14)
B:
(5,10) (14,14) (7,10)
(6,2) (1,9) (5,14)
(6,5) (12,12) (5,1)
C:
(-43,124) (227,52) (-4,172)
(-143,416) (48,114) (45,316)
(-38,224) (-131,182) (-163,94)
Обратная матрица C:
(3.2984e-05,-0.00299447) (-0.00238999,0.000217385) (0.00344114,-0.0023256)
(0.00508859,0.0033554) (-0.00374578,-0.00151086) (-0.000334049,0.00212238)
(-0.00423755,0.00372038) (0.0042166,-0.00152286) (-0.00576513,-0.000133187)
n_malder@WS-Maldera: /mnt/FILES/projects/нмсы на кладбище/tech-prog/third/third - [Вт мая 28, 10:29]
--($)-->git:(batain+)>
n_malder@WS-Maldera: /mnt/FILES/projects/нмсы на кладбище/tech-prog/third/third 10x53
n_malder@WS-Maldera: /mnt/FILES/projects/нмсы на кладбище/tech-prog/third/third - [Вт мая 28, 10:27]
--($)-->git:(batain+)> cat matrix_A.txt
6 2 1 12 7 6
14 1 12 4 7 3
7 8 7 2 3 14
n_malder@WS-Maldera: /mnt/FILES/projects/нмсы на кладбище/tech-prog/third/third - [Вт мая 28, 10:29]
--($)-->git:(batain+)> cat matrix_B.txt
5 10 14 14 7 10
6 2 1 9 5 14
6 5 12 12 5 1
n_malder@WS-Maldera: /mnt/FILES/projects/нмсы на кладбище/tech-prog/third/third - [Вт мая 28, 10:29]
--($)-->git:(batain+)>
```

3.4 Проверка через мат пакеты:

```
octave
octave 211x53
n_malder@WS-Maldera: /mnt/FILES/projects/нмсы на кладбище/tech-prog/third/third - [Вт мая 28, 10:57]
--($)-->git:(batain+)> octave
GNU Octave, version 6.2.0
Copyright (C) 1993-2023 The Octave Project Developers.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

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 = [6+2*i 1+12*i 7+6*i; 14+1*i 12+4*i 7+3*i; 7+8*i 7+2*i 3+14*i]
A =
   6 + 2i   1 + 12i   7 + 6i
  14 + 1i  12 + 4i   7 + 3i
   7 + 8i   7 + 2i   3 + 14i

octave:2> B = [5+10*i 14+14*i 7+10*i; 6+2*i 1+9*i 5+14*i; 6+5*i 12+12*i 5+1*i]
B =
   5 + 10i  14 + 14i   7 + 10i
   6 + 2i   1 + 9i   5 + 14i
   6 + 5i  12 + 12i   5 + 1i

octave:3> C = (A^2 + B^2)'
C =
 -43 + 124i  -143 + 416i  -38 + 224i
 227 + 52i   48 + 114i  -131 + 182i
 -4 + 172i  -131 + 182i  -163 + 94i

octave:4>
```