МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Забайкальский государственный университет» (ФГБОУ ВО «ЗабГУ»)

Факультет Энергетический

Кафедра Информатики, вычислительной техники и прикладной информатики

# ОТЧЕТ

по учебной практике

обучающегося Рычкова Родиона Викторовича

(фамилия, имя, отчество) Курс 1 Группа ВМК-21

Направления подготовки (специальности):

09.03.01 Информатика и вычислительная техника

(шифр, наименование)

Руководитель практики от кафедры

(Ученая степень, должность, Ф.И.О.)

г. Чита 2022

Оглавление

[ОТЧЕТ 1](#_Toc114751335)

Задание [11в 3](#_Toc114751336)

Задание [50………………………………………………………………………… 3](#_Toc114751337)

Задание [71а 4](#_Toc114751338)

Задание [78г 5](#_Toc114751339)

Задание [148 6](#_Toc114751340)

Задание [201б 7](#_Toc114751341)

Задание [260в 8](#_Toc114751342)

Задание [335в 9](#_Toc114751343)

Задание [388 10](#_Toc114751344)

Задание [425 11](#_Toc114751345)

Задание [471б 12](#_Toc114751346)

Задание [527 14](#_Toc114751347)

Задание [535 16](#_Toc114751348)

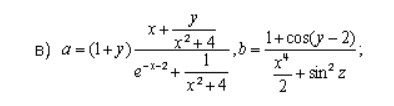
Задание [680б 18](#_Toc114751349)

Задание [708 20](#_Toc114751350)

Задание [846в 2](#_Toc114751351)3

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 2](#_Toc114751352)5

11 Даны x, y, z. Вычислить a, b, если



#include <cmath>

#include <iostream>

using namespace std;

int main()

{

float x, y, z, a, b;

cout << "Enter x, y, z\n" << endl; //Пользователь вводит значения для получения результата

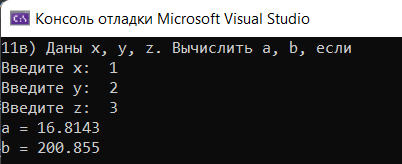
cin >> x >> y >> z; //Ввод значения

a = (1 + y)\*((x+(y/(x\*x+4)))/(exp(-x-2)+(1/(x\*x+4)))); //Ищем значение a из уравнения

b = (1 + cos(y - 2)) / ((pow(x, 4) / 2) + pow(sin(z), 2)); //Ищем значение b из уравнения

cout << "a = " << a << " b = " << b << endl; //Получение результата

}



**50**Даны действительные числа a1, b1, c1, a2, b2, c2. Выяснить, верно ли, что |a1b2 - a2b1| ≥ 0.0001, и если верно, то найти рашение системы линейных уравнений



#include <cmath>

#include <iostream>

using namespace std;

int main()

{

const double eps = 0.0001; //Значение экспоненты из задачи

double a1 = 0; //Значение переменной из задачи

double b1 = 0; //Значение переменной из задачи

double c1 = 0; //Значение переменной из задачи

double a2 = 0; //Значение переменной из задачи

double b2 = 0; //Значение переменной из задачи

double c2 = 0; //Значение переменной из задачи

cout << "Enter a1, b1, c1 for first equation" << endl; //Ввод пользователем значений переменных

cin >> a1 >> b1 >> c1; //Ввод

cout << "Enter a2, b2, c2 for second equation" << endl; //Ввод пользователем значений переменных

cin >> a2 >> b2 >> c2; //Ввод

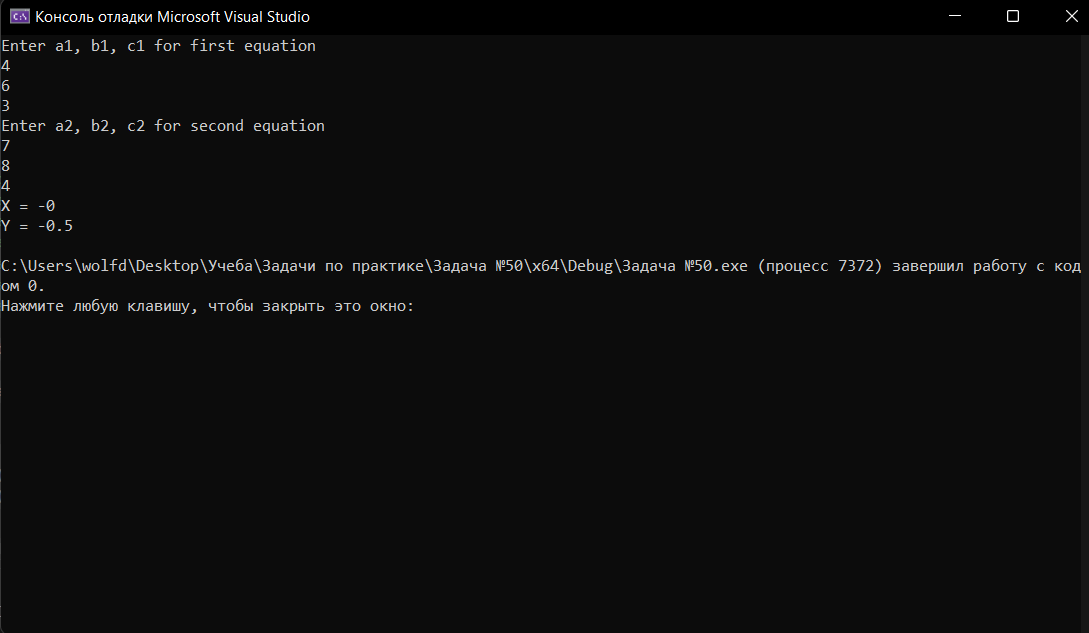
if (fabs(a1 \* b2 - a2 \* b1) >= eps) { //Условие для выполнения задачи

cout << "X = " << (b1 \* c2 - b2 \* c1) / (a1 \* b2 - a2 \* b1) << endl; //Вывод ответа

cout << "Y = " << (c1 \* a2 - c2 \* a1) / (a1 \* b2 - a2 \* b1) << endl; //Вывод ответа

}

}



**71**Дано действительное число *а.*Вычислить *f(a),*где*f* — периодическая функция с периодом 1.5, совпадающая на отрезке [0, 1.5]: а) с функцией *х*3— 2.25*x*;

#include <cmath>

#include <iostream>

using namespace std;

int main()

{

float f, a, b;

double p = 1.5;

int c;

cout << "Enter a: "; //Ввод пользователем значения a

cin >> a; //Ввод

if (a >= 0) { //Условие при котором выполняется задача

c = a / p;

b = a / p;

a = (b - c) \* p;

}

else if (a < 0) { //Иначе если условие неверно

c = (-a) / p;

a = (c + 1) \* p + a;

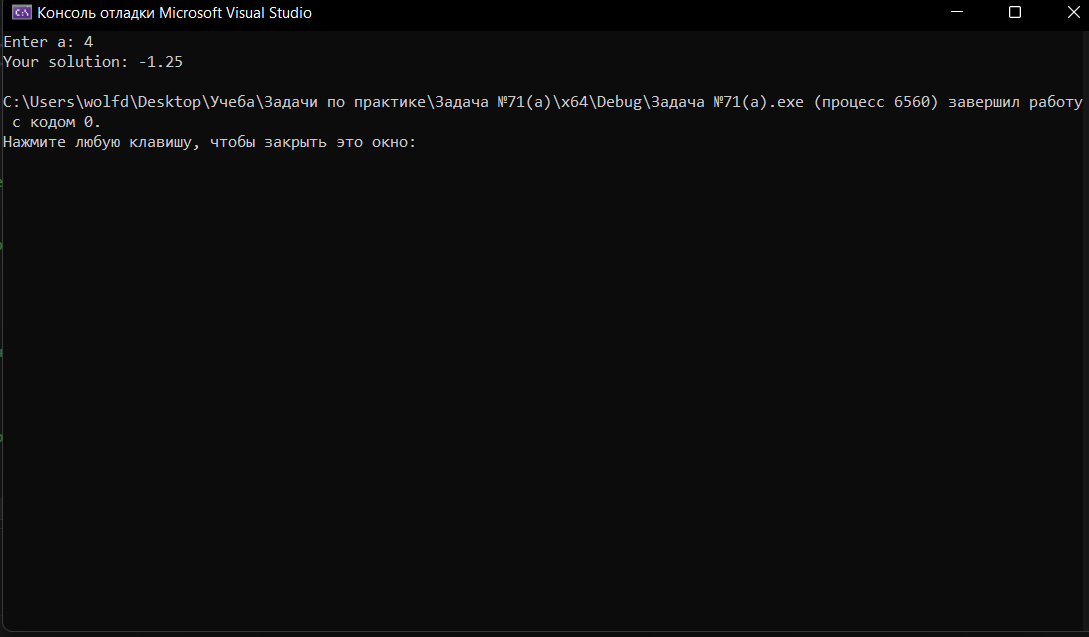
}

f = a \* a \* a - 2.25 \* a; //Нахождение результата задачи

cout << "Your solution: " << f << endl; //Вывод ответа

return 0;

}



**78**Даны действительное число *а*, натуральное число n. Вычислить:

г) 

#include <cmath>

#include <iostream>

using namespace std;

int main()

{

int a, n, f;

double sum;

cout << "Enter final number: "; //Ввод последнего числа

cin >> f; //Ввод

cout << "Enter number a: "; //Ввод значения переменной

cin >> a; //Ввод

sum = 1.0 / a;

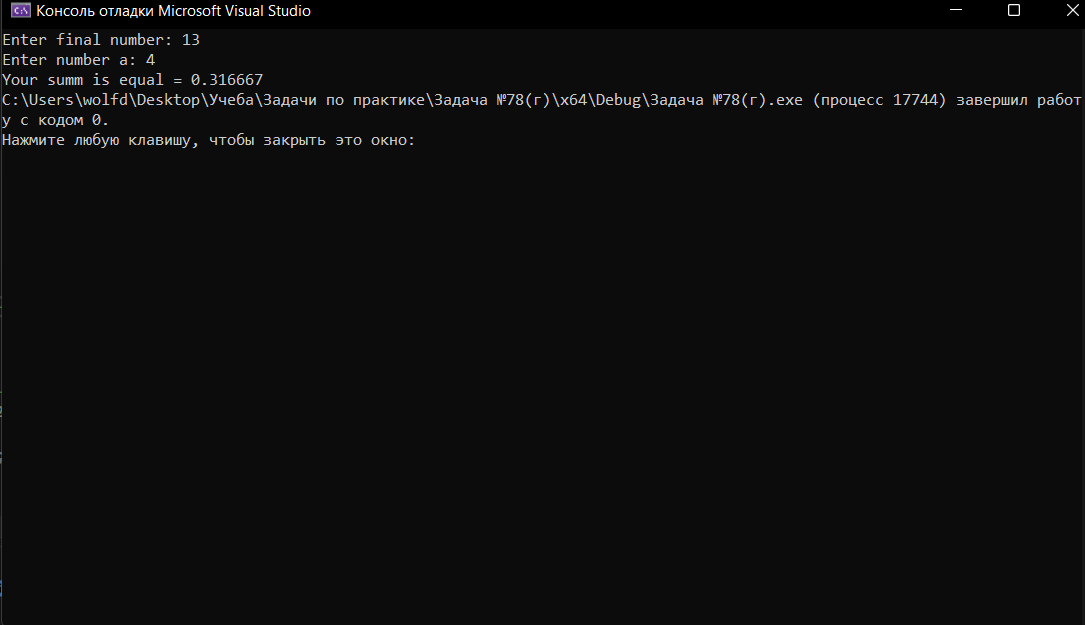
for (n = 1; n <= f; n++) { //Цикл для нахождения ответа

sum = sum + 1 / pow(a, 2 \* n); //Решение задачи

}

cout << "Your summ is equal = " << sum; //Вывод результата

}



**148**Получить таблицу температур по Цельсию от 0 до 100 градусов и их эквивалентов по шкале Фаренгейта, используя для перевода формулу tF = (9/5)tc + 32.

#include <iostream>

using namespace std;

int i = 0;

typedef struct ListStruct { // Односвязный список

int x;

struct ListStruct\* next;

} my\_list;

void push\_end(my\_list\* h, my\_list\* n) { // Добавить в конец списка

my\_list\* act = h;

while (act->next != nullptr) {

act = act->next;

}

act->next = n;

}

void new\_list(my\_list\* h) { // Создать элемент списка

my\_list\* next = new my\_list;

next->x = i;

next->next = nullptr;

push\_end(h, next);

}

void view\_list(my\_list\* h) { // Вывести список на экран

my\_list\* act = h;

while (act != nullptr) {

cout << act->x << " ";

act = act->next;

}

cout << endl;

}

int temper(my\_list\* h) {

int temp = 0;

my\_list\* act = h;

while (act != nullptr)

{

temp = (1.8 \* act->x) + 32; // tF = (9 / 5)tc + 32.

cout << "Temperature C=" << act->x << " —> F=" << temp << endl;

act = act->next;

}

return temp;

}

void delete\_list(my\_list\* h) { // Удалить список

my\_list\* act;

while (h != nullptr) {

act = h;

h = h->next;

act->next = nullptr;

free(act);

}

}

int main() {

my\_list\* head = new my\_list;

head->x = 0;

head->next = nullptr;

for (i = 1; i < 101; i++) {

new\_list(head); // собираю цельсии

}

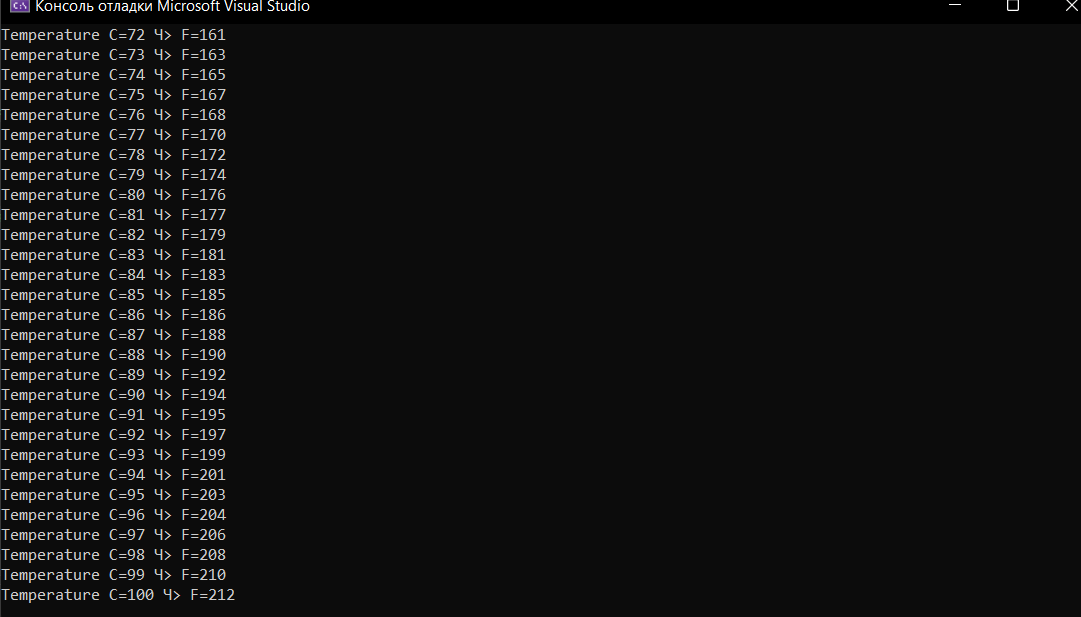
//view\_list(head);

temper(head); // перевод из цельсия в фаренгейты

delete\_list(head);

return 0;

}



**201**Даны натуральное число n, действительные числа a1,...,an. Получить:

б) min (a1,...,an);

#include <iostream>

#include <cstdlib>

using namespace std;

const int n = 10;

int main()

{

int mass[n], min; //Создание массива

cout << "Elements: |"; //Получение списка массива

for (int i = 0; i < n; i++) //Заполнение массива рандомными значениями

{

mass[i] = rand()%99; //Получение небольших чисел

cout << mass[i] << "|"; //Вывод массива

}

cout << endl;

min = mass[0];

for (int i = 1; i < n; i++) //Цикл для решение задачи

{

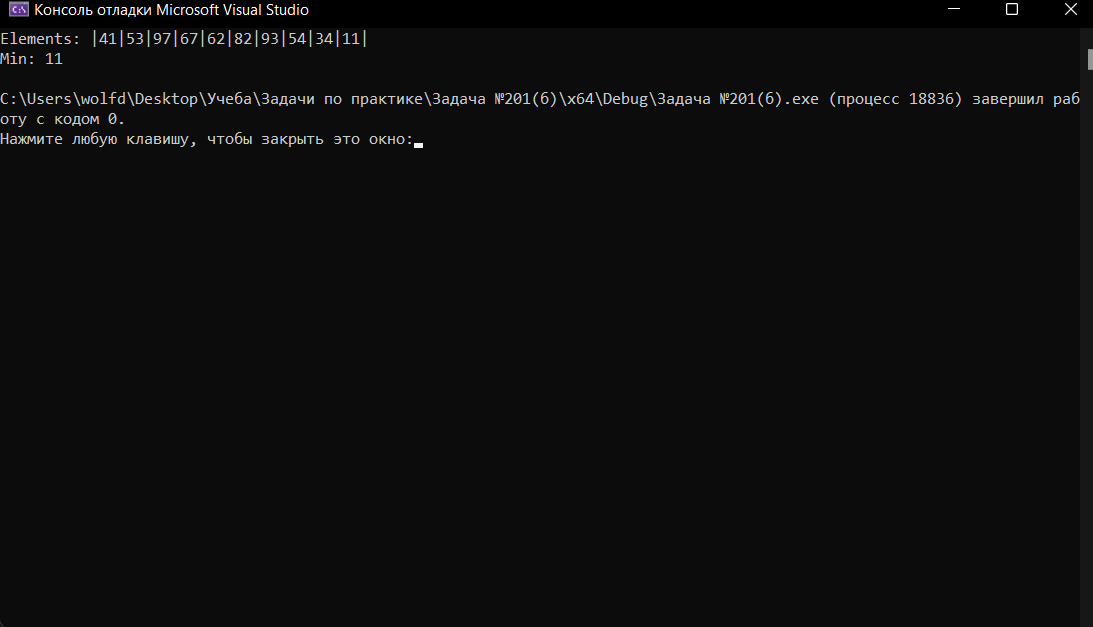
if (min > mass[i]) min = mass[i];//Условие для решения задачи

}

cout << "Min: " << min << endl; //Вывод минимального значения из массива

return 0;

}



**260**Даны натьуральное число *n*, и символы *s1*,..., sn, среди которых есть двоеточие.

в) Получить все символы, расположенные между первым и вторым двоеточием. Если второго двоеточия нет, то получить все символы, расположенные после единственного имеющегося двоеточия.

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s\_1, s\_2;//Строки

size\_t i = 0;//целочисленный тип без знака, который гарантированно будет достаточно большим, чтобы хранить размер в байтах любого объекта в памяти.

getline(cin, s\_1);//Осуществляем ввод строки пользователем

while (s\_1[i] != ':')//Делаем счетчик до нахождения первого двоеточия

i++;//Увеличиваем счетчик на 1

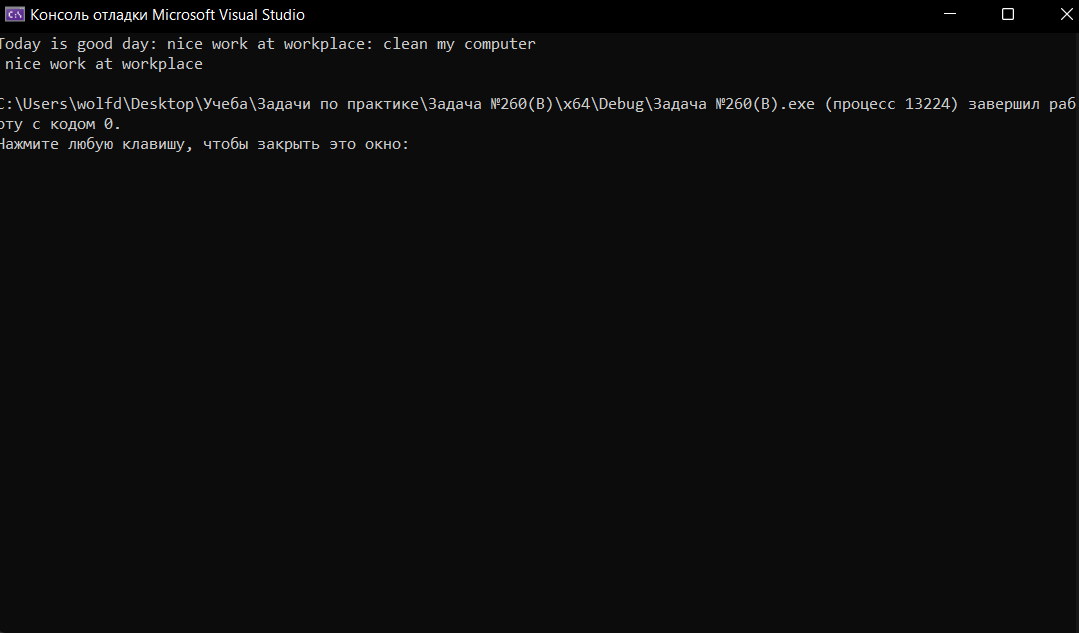
while (s\_1[++i] != ':' && s\_1[i] != '\0')//Пока счетчик не равен двоеточию и не равен концу строки

s\_2 += s\_1[i];//Выводим слова после двоеточия

cout << s\_2 << endl;

return 0;

}



**335**Дано натуральное число n. Вычислить:

в);

//335 Дано натуральное число n.Вычислить:

#include <iostream>

#include <math.h>

using namespace std;

int fact(int p) {

int q = 1;

for (int i = p - 1; i > 0; i--) {

p = p \* i;

}

cout << p << endl;//Проверка факториала

return p;

}

int main()

{

int n;

double sum = 0.0;

cin >> n;

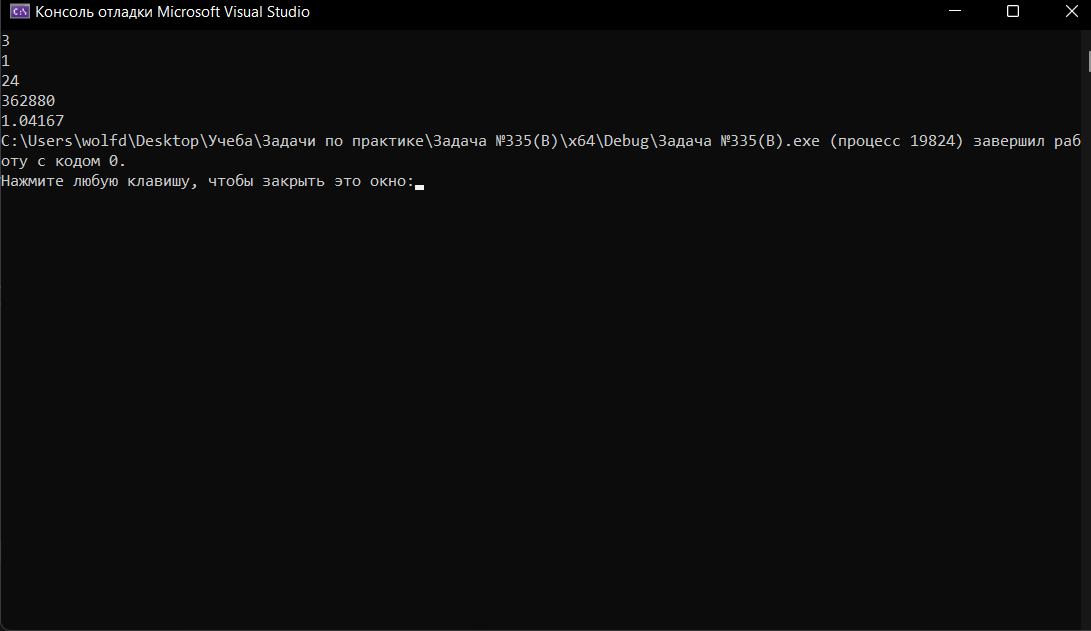
for (int k = 1; k <= n; k++) {

sum = sum + 1.0 / (fact(pow(k, 2)));

}

cout << sum;

}



**388**В данной квадратной целочисленной матрице порядка 17 указать индексы всех элементов с наибольшим значением.

#include "iostream"

#include "ctime"

#include "cstdlib"

#include "iomanip"

#include "climits"

using namespace std;

int main() {

const size\_t n = 17; //Размер матрицы

int a[n][n], m = INT\_MIN;//Минимальное значение из матрицы

srand(time(NULL)); //Генерируем случайные числа

for (size\_t i = 0; i < n; i++) { //Цикл i-той строки

for (size\_t j = 0; j < n; j++) //Цикл j-той строки

cout << setw(5) << (a[i][j] = rand() % 199 - 99); //Выводим случайные числа в матрице

cout << endl;//Придаем вид матрице

}

for (size\_t i = 0; i < n; i++)//Цикл i-той строки

for (size\_t j = 0; j < n; j++)//Цикл j-той строки

if (a[i][j] > m && i != j && i + j + 1 != n)m = a[i][j];//Условие для получения маскимального значения из матрицы

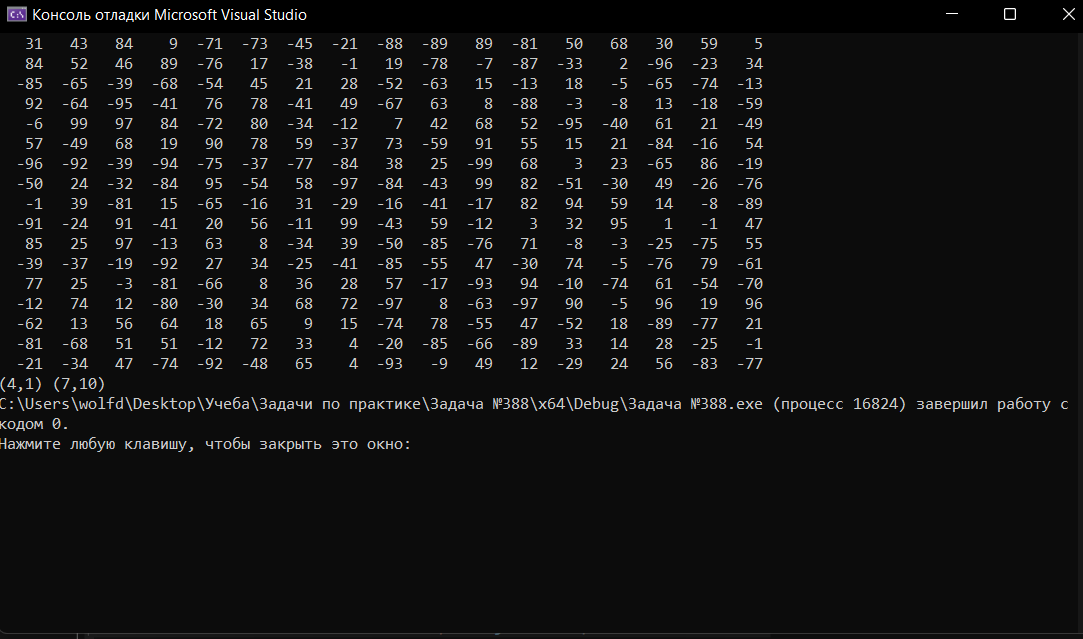
for (size\_t i = 0; i < n; i++)//Цикл i-той строки

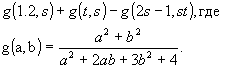
for (size\_t j = 0; j < n; j++)//Цикл j-той строки

if (i != j && i + j + 1 != n && m == a[i][j]) //Получаем индексы максимальных элементов

cout << "(" << i << ',' << j << ") "; //Выводим индексы

}



**425**Даны действительные числа s, t. Получить  


#include <math.h>

#include <iostream>

using namespace std;

double g(int a, int b) {

return (pow(a, 2) + pow(b, 2)) / pow(a, 2) + (2 \* a \* b) + (3 \* pow(b, 2)) + 4; //Вычисления по формуле из условия задачи

}

int enter() { //Функция ввода значений переменных

int s, t;

int solution;

cout << "Enter parametres: " << endl; //Ввод параметров

cin >> s;

cin >> t;

cout << "You entered: " << s << ", " << t << endl; //Вывод введенных параметров

solution = g(1.2, s) + g(t, s) - g(2 \* s - 1, s \* t);//Главное решение задачи

cout << "Your solution: " << solution << endl;//Вывод ответа

return g(s, t);

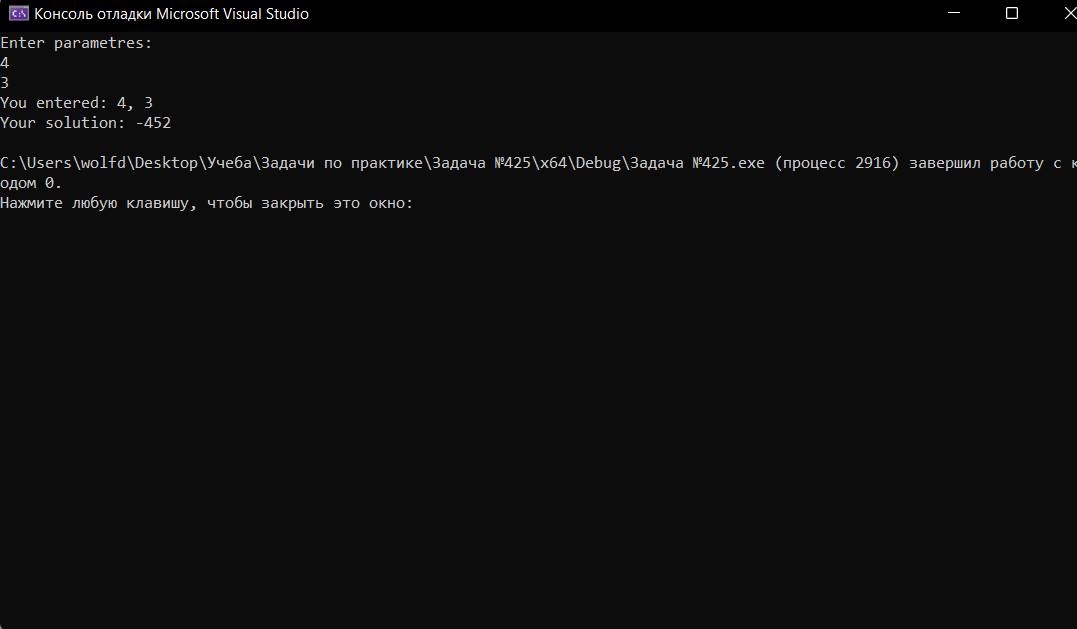
}

int main()

{

enter();

}



**471**Дан файл f, компоненты которого являются действительными числами. Найти: б) произвидение компонент файла f;

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ifstream f("f.txt", ifstream::in);//Открываем файл для чтения

double v = 1, s = 1;

while (!f.eof()) { //Пока не найден конец файла

f >> v;//Оператор сдвига влево

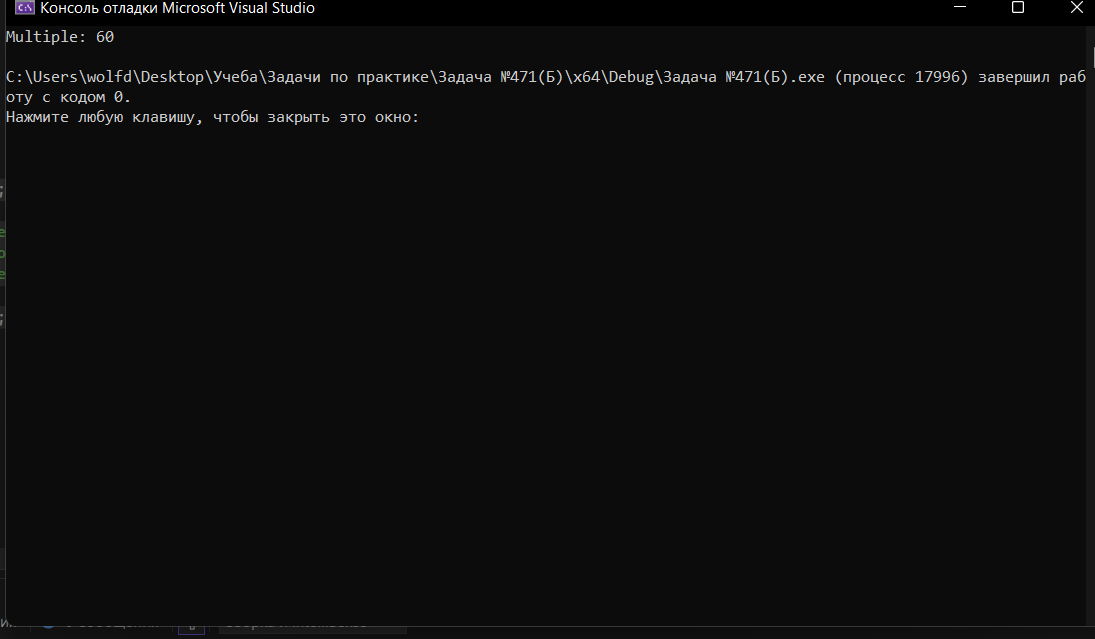
s \*= v;//Умножаем произведение

}

cout << "Multiple: " << s << endl; //Выводим результат

return 0;

}





**527**Даны два текстовых файла f и g. Определить, совпадают ли компоненты файла f с компонентами файла g. Если нет, то получить номер первой строки и позицию первого символа в этой строке, в кототых файлы f и g отличаются между собой. Принять во внимание уточнение к задаче [494](https://ivtipm.github.io/Programming/Glava13/index13.htm#z494).

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <locale.h>

int n = 0, k = 0, str = 1, num = 1;

char s, s2;

FILE\* f, \* f2;

char name[] = "f.txt";

char name2[] = "g.txt";

int main()

{

setlocale(LC\_ALL, "Rus");

f = fopen(name, "r");

f2 = fopen(name2, "r");

fseek(f, 0, SEEK\_END);//Переводим указатель на конец строки

int size1 = ftell(f);

fseek(f2, 0, SEEK\_END);//Переводим указатель на конец строки

int size2 = ftell(f2);

if (size1 < size2)

{

fseek(f2, 0, SEEK\_SET);

fseek(f, 0, SEEK\_SET);

while (!feof(f2))//Пока не равно концу файла

{

s = fgetc(f);//получает следующий по порядку символ из входного потока

s2 = fgetc(f2);//получает следующий по порядку символ из входного потока

if (s == s2) n++;

k++;

if (s2 == '\n')

{

str++; n = 0; k = 0;

}

}

}

if (size1 >= size2)

{

fseek(f, 0, SEEK\_SET);

fseek(f2, 0, SEEK\_SET);

while (!feof(f))

{

s = fgetc(f);

s2 = fgetc(f2);

if (s == s2) n++;

k++;

if (s == '\n')

{

str++; n = 0; k = 0;

}

}

}

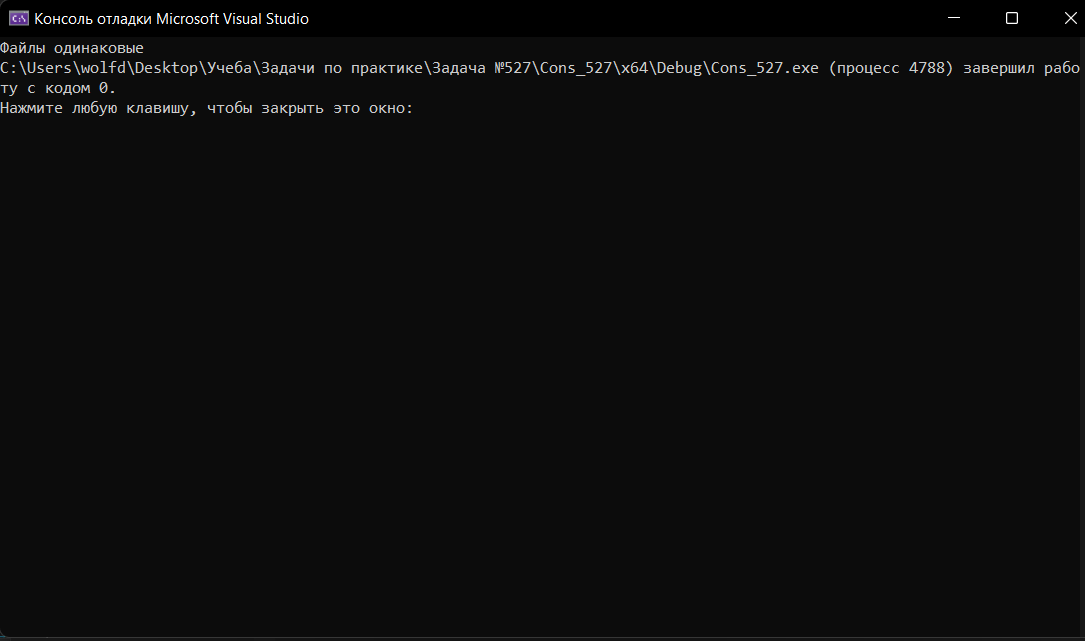
if (n == k) printf\_s("Файлы одинаковые");

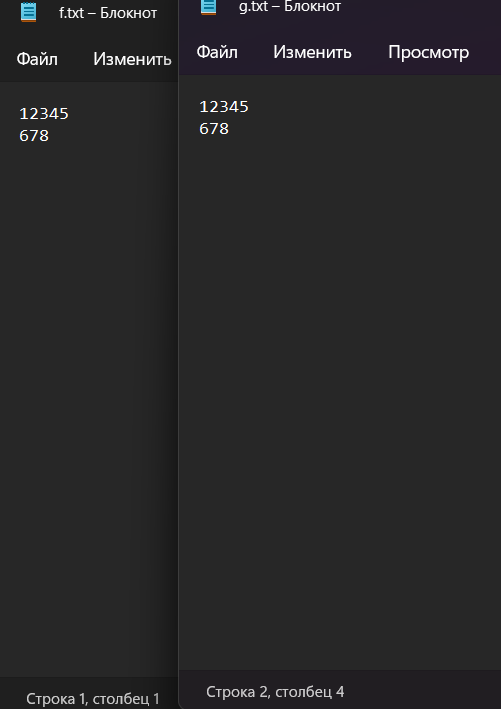
else printf\_s("Первый не совпадающий символ находится в строке %d под номером %d", str, n);

fclose(f);

fclose(f2);

}





535 Пусть a1 = 1; a2 = 1.5; ai = a[i/2]a[i/3] + 1 ( i = 3, 4...). Дано натуральное число m. Получить am.

#include <iostream>

#include <locale>

using namespace std;

// структура узла

struct Node

{

double u = 0; // число

Node\* next = nullptr; // указатель на следующий узел

Node\* prev = nullptr; // указатель на предыдущий узел

};

// возвращает a[i] - смотри условия задачи

double number(Node\* head, int n)

{

Node\* node = head;

for (int i = 1; i < n; i++)

node = node->next;

return node->u;

}

// создание списка. Функция возвращает указатель на последний элемент

Node\* build\_list(Node\* head, Node\* tail, int n)

{

head->u = 1;

for (int i = 2; i <= n; i++)

{

Node\* node = new Node();

node->prev = tail;

tail->next = node;

if (i == 2)

node->u = 1.5;

else

node->u = number(head, i / 2) \* number(head, i / 3) + 1;

tail = node;

}

return tail;

}

// вывод спсика в консоль

void print\_list(Node\* head)

{

Node\* dop\_node = head;

while (dop\_node)

{

cout << dop\_node->u << " ";

dop\_node = dop\_node->next;

}

}

// вывод |a1|, |a1 + a2|, ..., |a1 + a2 + ... + an|

void out\_result(Node\* head)

{

Node\* node = head;

double res = 0;

while (node)

{

res += abs(node->u);

cout << res << " ";

node = node->next;

}

}

// удаление спсика

void delete\_list(Node\* head)

{

Node\* dop\_node;

while (head)

{

dop\_node = head;

head = head->next;

delete dop\_node;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

cout << "535 Пусть a[1] = 1; a[2] = 1.5; a[i] = a[i/2]a[i/3] + 1 ( i = 3, 4...). Дано натуральное число m. Получить a[m]." << endl;

int m;

cout << "Введите n = "; cin >> m;

if (m > 0)

{

Node\* head; // первый элемент списка

Node\* tail; // последний элемент списка

head = tail = new Node();

// создание списка

tail = build\_list(head, tail, m);

cout << endl << "Действительные числа а1,..., am" << endl;

print\_list(head); // вывод списка

cout << endl << "Полученное a[m] = " << tail->u << endl;

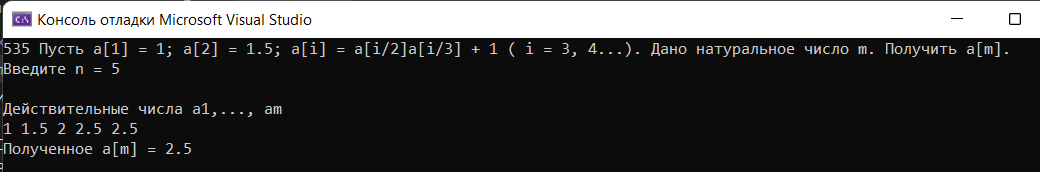
delete\_list(head); // удаление списка

}

else

cout << " Error -- m < 1 " << endl;

}



**680**В данной действительности матрице размера *n x m* (n ≥ 3, m ≥ 3) поменять местами: б) столбцы с номерами 3 и *n-2*

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <iomanip>

using namespace std;

void task(int\*\* array, int x) {

int i;

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

{

int temp = array[i][2];

array[i][2] = array[i][x - 3];

array[i][x - 3] = temp;

}

}

void matrix\_output(int\*\* array, int x, int y) {

int i, j;

for (i = 0; i < x; i++) // выводим отсортированную

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

{

if (j % y == 0)

cout << endl;

cout << setw(3) << array[i][j];

}

cout << endl;

}

int foo\_input\_output(int x) {

cout << "Print count line " << endl;

cin >> x;

return(x);

}

int foo\_input\_output1(int y) {

cout << "Print count columns " << endl;

cin >> y;

return(y);

}

void matrix\_output1(int\*\* array, int x, int y) {

int i, j;

for (i = 0; i < x; i++) // выводим исходную матрицу

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

{

if (j % y == 0)

cout << endl;

cout << setw(3) << array[i][j]; // setw = изменение ширины поля ввода-вывода

}

cout << endl;

}

int matrix\_input(int\*\* array, int x, int y) {

int i, j;

cout << "Enter matrix " << endl;

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

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

cin >> array[i][j];

return \*\*array;

}

void main()

{

int x = 0, y = 0;

int i, j;

x = foo\_input\_output(x); // вводим n

y = foo\_input\_output1(y); //вводим m

int\*\* array; // динамически выделяем память

array = new int\* [x];

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

array[i] = new int[y];

matrix\_input(array, x, y); // вводим матрицу

matrix\_output1(array, x, y); // выводим ориг матрицу

task(array, x); // выполняем задание

matrix\_output(array, x, y); // выводим матрицу по заданию

delete array; // освобождаем память

}



**708**Даны квадратная матрица А порядка m, натуральное число n. Получить матрицу *E + A + A2 + ... + An*, где Е - единичная матрица порядка m.

#include <iostream>

#include <locale>

using namespace std;

// функция по заполнению квадратной матрицы mat порядка m

void fill\_mat(double\*\* mat, int m)

{

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

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

mat[i][j] = rand() % 10;

}

// создаёт единичную матрицу порядка m

double\*\* matrix\_one(int m)

{

// выделяем память под матрицу

double\*\* mat = new double\* [m];

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

mat[i] = new double[m];

// заполняем её

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

{

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

{

if (i != j)

mat[i][j] = 0;

else

mat[i][j] = 1;

}

}

// возвращаем единичную матрицу (указатель на неё)

return mat;

}

// вывод квадратной матрицы

void print\_mat(double\*\* mat, int m)

{

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

{

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

cout << mat[i][j] << "\t"; // "\t" - это табуляция

cout << endl;

}

}

// функция по выделению памяти под квадратную матрицу порядка m

double\*\* build\_matrix(int m)

{

double\*\* mat = new double\* [m];

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

mat[i] = new double[m];

return mat;

}

// очищение памяти от двумерного массива

void delete\_matrix(double\*\* mat, int m)

{

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

delete[] mat[i];

delete[] mat;

}

// перемножение квадратной матрицы "a" самой на себя

double\*\* multi(double\*\* a, int m, double\*\* b)

{

// выделение памяти под матрицу, которая будет хранить в себе результат

double\*\* res = new double\* [m];

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

res[i] = new double[m];

// перемжножение

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

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

{

res[i][j] = 0;

for (int k = 0; k < m; k++)

res[i][j] += a[i][k] \* b[k][j];

}

// удаление старой матрицы

delete\_matrix(a, m);

// возвращение результата, произведения A\*A

return res;

}

// сложение матриц a и b, результат сохраняется в a

void summa\_matrix(double\*\* a, double\*\* b, int m)

{

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

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

a[i][j] = a[i][j] + b[i][j];

}

int main()

{

setlocale(LC\_ALL, "rus");

cout << "708 Даны квадратная матрица А порядка m, натуральное число n." << endl;

cout << "Получить матрицу E + A + A^2 + ... + A^n, где Е - единичная матрица порядка m." << endl;

int n, m;

cout << "Введите n: "; cin >> n;

if (n > 0)

{

cout << "Введите порядок квадратной матрицы (m >= 1): ";

cin >> m;

cout << endl;

// Создаём единичную матрицу

double\*\* E = matrix\_one(m);

// и выводим её в консоль

cout << "Единичная матрица порядка m:" << endl;

print\_mat(E, m);

cout << endl;

// создаём двумерный динамический массив - матрицу А размером m\*m

double\*\* A;

A = build\_matrix(m);

// заполнение матрицы А

fill\_mat(A, m);

// вывод матрицы А

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

print\_mat(A, m);

cout << endl;

// дополнительная матрица

double\*\* dop = build\_matrix(m);

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

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

dop[i][j] = A[i][j];

// вычисляем E + A + A^2 + ... + A^n

summa\_matrix(E, A, m); // тут находим сумму E + A

// а в цикле уже остальные суммы

for (int i = 1; i < n; i++)

{

A = multi(A, m, dop);

summa\_matrix(E, A, m);

}

cout << "Результат -- E + A + A^2 + ... + A^n" << endl;

print\_mat(E, m);

// очищение памяти от двумерных массивов

delete\_matrix(A, m);

delete\_matrix(dop, m);

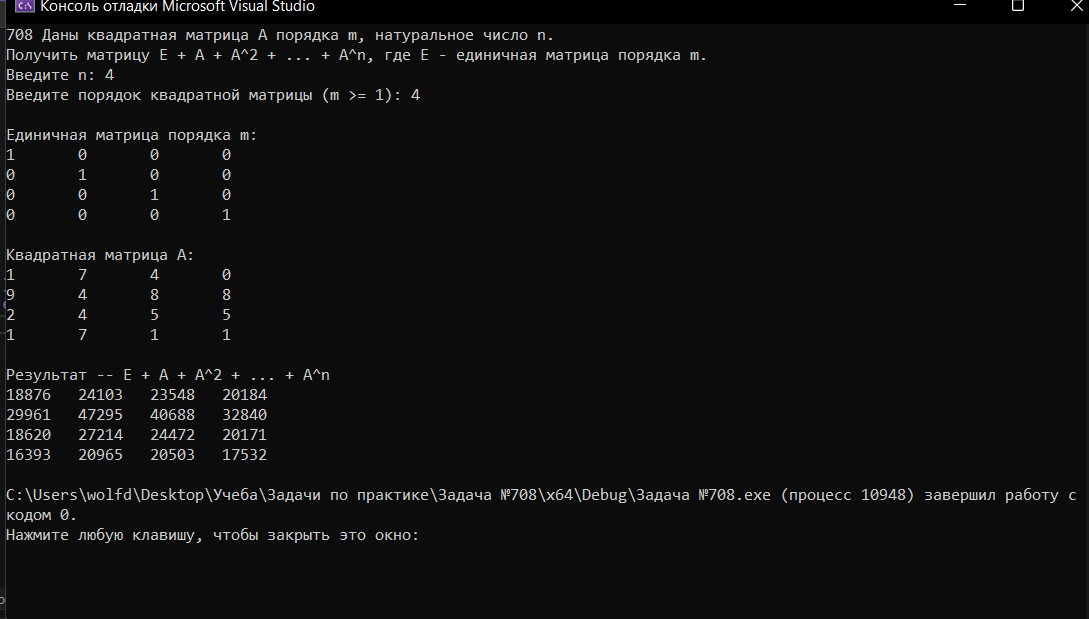
delete\_matrix(E, m);

}

else

cout << "n не натуральное число, т.е. n < 1" << endl;

}



846 Исследовать область определения и построить графики следующих функций:



#include <iostream>

#include <locale>

#include <Windows.h> // для рисования

#include <conio.h> // для рисования

using namespace std;

double f(double x)

{

return 3 + (2 / x) + (3 / (x \* x));

}

int main()

{

setlocale(LC\_ALL, "rus");

POINT op;

HWND hwnd = GetConsoleWindow();

HDC hdc = GetDC(hwnd);

SelectObject(hdc, GetStockObject(WHITE\_PEN));

// рисуем ось x

MoveToEx(hdc, 0, 400, &op);

LineTo(hdc, 1000, 400);

// рисуем ось y

MoveToEx(hdc, 500, 0, &op);

LineTo(hdc, 500, 800);

double y, x;

// рисуем график

for (x = -20; x <= 20; x += 0.001)

{

if (x != 0)

{

SetPixel(hdc, x \* 20 + 500, -f(x) \* 20 + 400, RGB(255, 255, 255)); // рисуем по пиксельно каждую точку функции (масштаб отличается от реального в 20 раз)

}

}

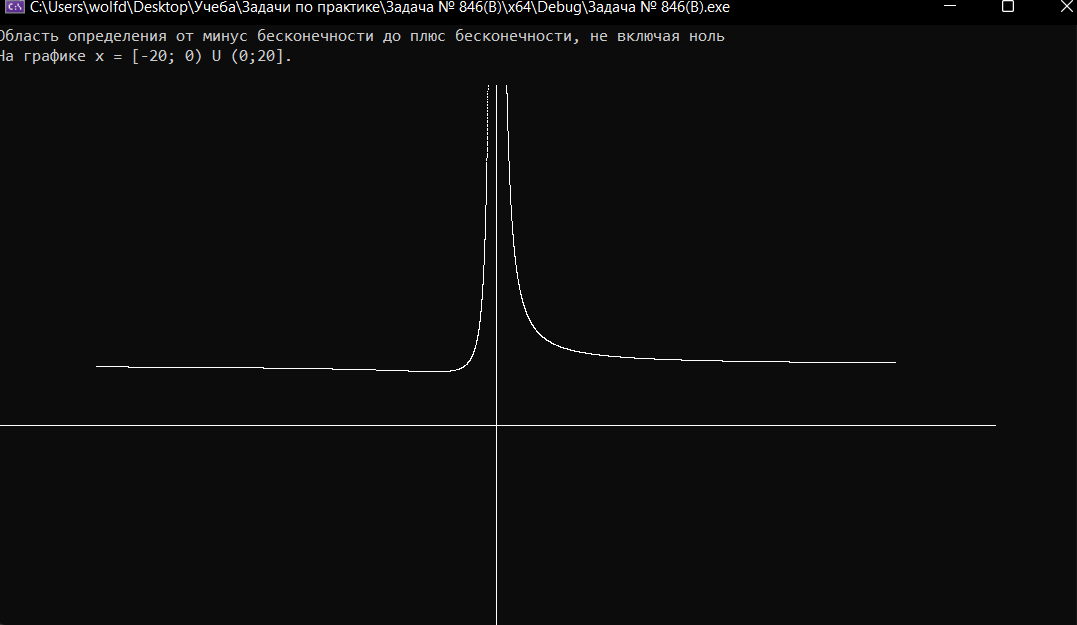
cout << "Область определения от минус бесконечности до плюс бесконечности, не включая ноль" << endl;

cout << "На графике x = [-20; 0) U (0;20]." << endl;

ReleaseDC(hwnd, hdc);

cin.get();

return 0;

}

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

<https://proglib.io/p/cpp-lessons>

<https://metanit.com/cpp/tutorial/>

<http://cppstudio.com/post/389/>

<https://codelessons.ru/cplusplus/ukazateli-v-c-podrobnoe-rukovodstvo.html>

<http://cppstudio.com/post/446/>

<https://metanit.com/cpp/tutorial/2.10.php>

<https://progtips.ru/biblioteki/13-bibliotek-s-o-kotoryx-nuzhno-znat.html>