**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**Воронежский государственный технический университет**

**Факультет информационных технологий и компьютерной безопасности**

**Кафедра искусственного интеллекта и цифровых технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

По дисциплине: Объектно-ориентированное программирование

Авторы работы: ­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­­ А. О. Анохин

подпись, дата инициалы, фамилия

Жаий Руфеида

подпись, дата инициалы, фамилия

группа бТИИ-241

Направление подготовки: 09.03.02 «Информационные системы и технологии»

Руководитель: \_\_\_\_\_\_\_\_\_­­­\_\_\_\_\_\_\_\_\_ доц. С. Г. Корчагин

подпись, дата должность, инициалы, фамилия

Воронеж

2025

# Постановка задачи

Задание 1

Найти значение алгебраического выражения, соответствующего варианту задания. Вывести результаты на печать. Все результаты выводить в развёрнутом виде (например: «Сумма чисел А и B равна 3.7854»).

Значения вводимых величин должны иметь не менее четырех значащих цифр и задаются студентом самостоятельно.

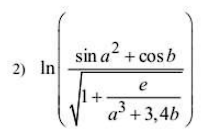


Рисунок 1 — второй вариант задания 1

Задание 2

Составить программу вычисления функции F(x), разложенной в ряд, сходящийся в заданной области.

Вычисления проводить до тех пор, пока модуль разности между последующим и предыдущим членами ряда не будет меньше или равен ε. Значение ε задается константой и для всех вариантов составляет 0,001.

Решить задачу для различных значений х. При решении задач массивы не использовать. Специальные функции для возведения в степень не использовать.

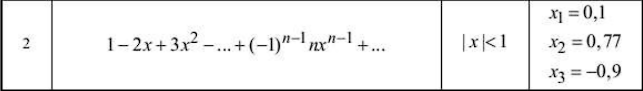


Рисунок 2 — второй вариант задания 2

Задание 3

В задании всего 2 варианта: вариант 1 выполняют те, у кого основной вариант нечетный, вариант 2 выполняют те, у кого основной вариант четный.

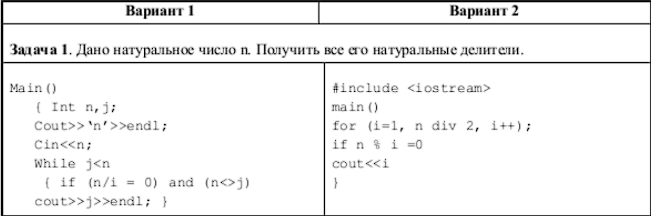
Каждый вариант содержит 15 задач, содержащих те или иные ошибки (логические, ошибки синтаксиса и т.п.). Задача состоит в следующем: используя средства отладчика среды, обнаружить все ошибки и сделать программу работоспособной (т.е. выполняющей то, что от нее требуется).

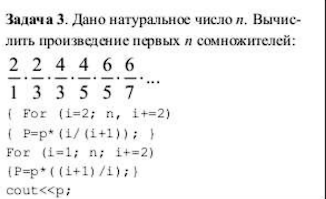
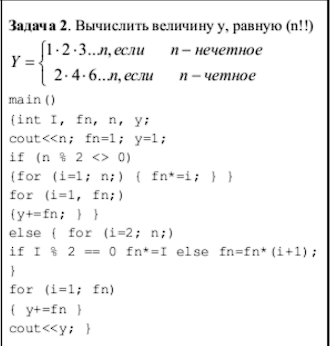
Для отчета по данному заданию необходимо:

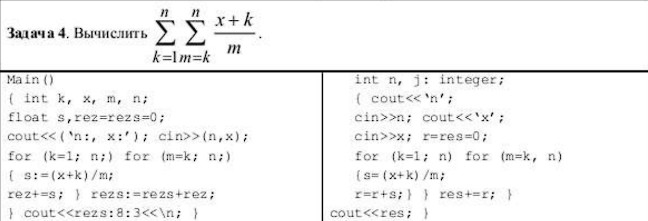
- показать работающие программы;

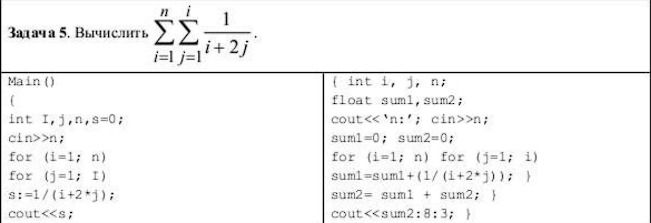
- код программ должен быть структурированным;

- к каждой программе необходимо составить список обнаруженных ошибок (согласно сообщениям отладчика), а также пояснения того, как вы исправляли эти ошибки.









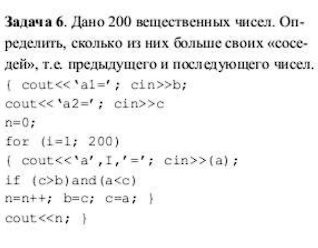


Рисунок 3 — задачи задания 3

Задание 4



Рисунок 4 — второй вариант задания 4

С помощью оператора выбора реализовать меню (возможность выбора номера нужной задачи для ее проверки преподавателем).

# Отчёт по заданию 1

В задании необходимо вычислить функцию (рисунок 1) с задаваемыми пользователем значениями a и b. Файл <cmath> использоваться не будет, функции корня, натурального логарифма, косинуса и синуса будут расписаны самостоятельно.

Функция корня.

Функция корня работает по принципу поиска ближайшего целого числа к заданному, имеющего целый квадратный корень. Затем ищется среднее арифметическое от суммы заданного числа, поделённого на ближайшее, и ближайшего числа. Затем вместо ближайшего числа используется получившееся арифметическое и т.д. Чем больше подобных усреднений будет выполнено, тем точнее получится результат. В данном случае достаточно трёх (код см. в приложении).

Функции синуса и косинуса.

Сначала функции подводят угол к значению <360 (если пользователь ввёл больший). Затем высчитывается знак синуса или косинуса. Далее угол уменьшается до <90. В функциях содержится массив со значениями всех косинусов или синусов для углов, кратных 10 (10, 20 ... 90). Функции высчитывают, к какой десятке относится угол (например угол 46 относится к 40). Функция прибавляет (или вычитает для косинуса) к найденной десятке единицы угла, равные разности этой и следующей десяток, деленных на 10 (шаг 1 градуса), умноженной на количество единиц угла (код см. в приложении).

Функция натурального логарифма.

После проверки, больше ли 0 введённое число, функция делит его на E = 2.718281828459045235, пока число больше единицы, и умножает его на E, пока оно меньше .25. При каждом делении целая переменная pa увеличивается на 1, при каждом умножении убавляется на 1. Затем число уменьшается на 1 и проводится обратный поиск степени (код см. в приложении).

Каждая функция принимает число типа double и возвращает число типа double. Остаётся только добавить в программу введение пользователем значений a и b, вычисление функции с помощью созданных функций и вывод результата на экран.

Проверим программу, введя числа a = 3.4567, b = 56.787.

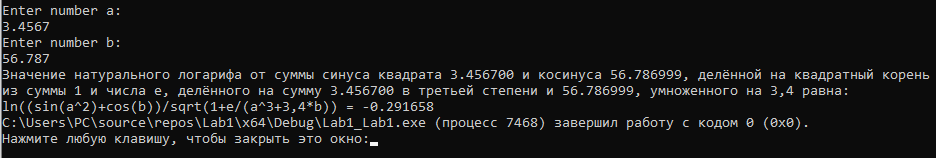


Рисунок 6 — тест программы 1

Теперь проверим результат.

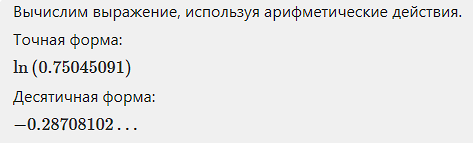


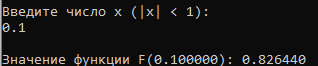
Рисунок 7 — проверка результата 1

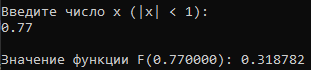
Как видно, программа довольно точно вычислила значение функции.

# Отчёт по заданию 2

Второе задание связано с вычисление ряда при заданном значении x. Оно будет записано в void функции task2. Суть проста: выполняется цикл с проверкой, больше ли разность двух последних значений суммы 0.001, в котором вычисляется слагаемое ряда, расписанное с помощью математических операций и функций.

Проверим программу, введя числа 0.1, 0.77 и -0.9.





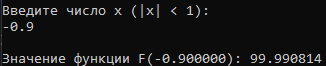
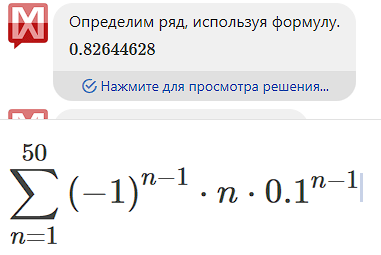
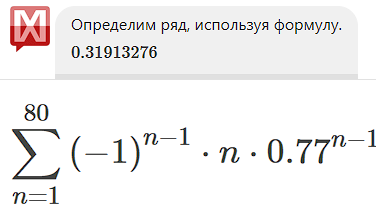


Рисунок 8 — тест программы 2

Теперь проверим получившиеся результаты.





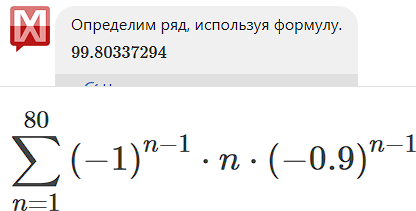
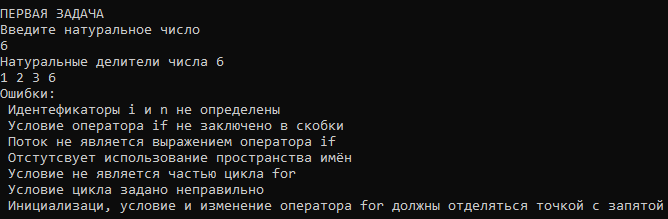


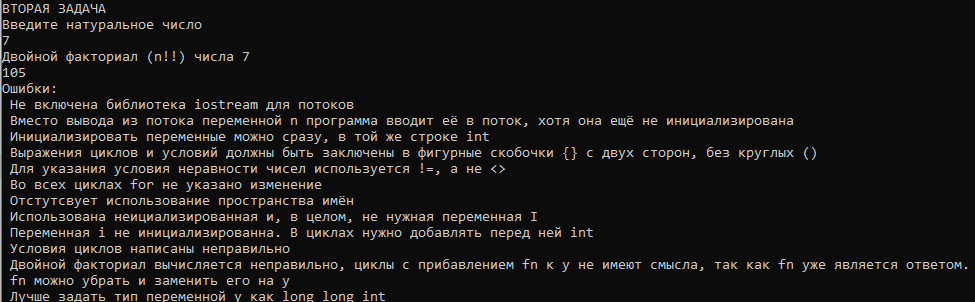
Рисунок 9 — проверка результатов 2

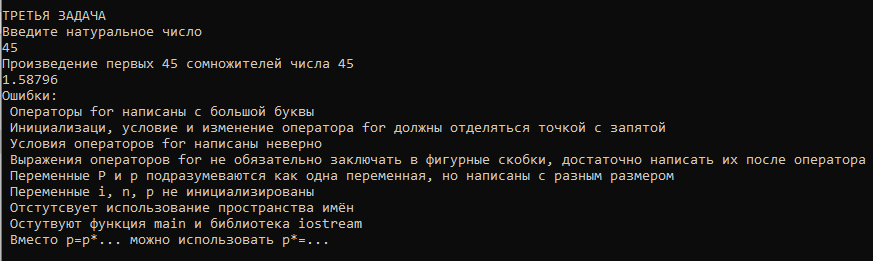
Как видно, программа весьма точно вычисляет значение суммы ряда.

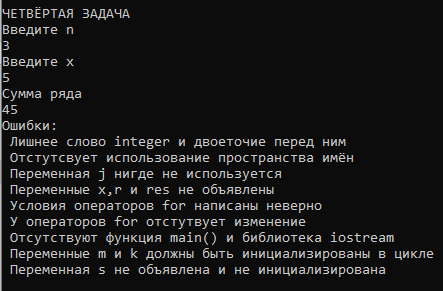
# Отчёт по заданию 3

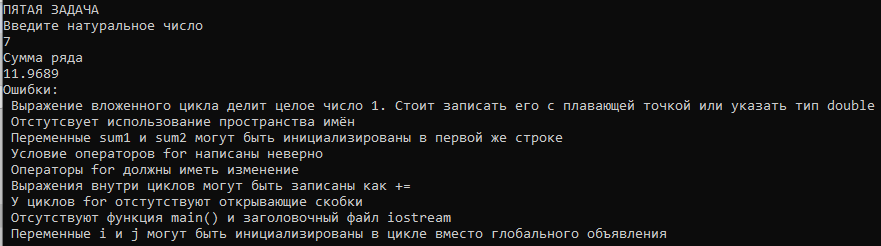
Третье задания состоит из исправлении ошибок показанных программ. Каждая из семи программ будет записана в отдельной функции в уже исправленном виде. Список ошибок будет выводится на экран.

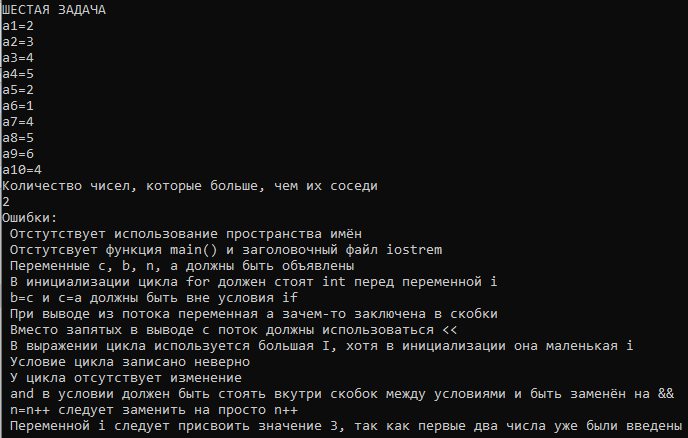












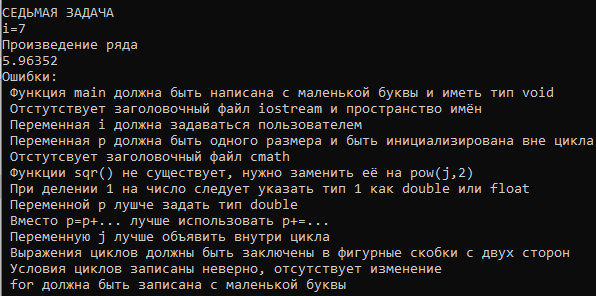


Рисунок 10 — результат работы программ по задаче 3

# Отчёт по заданию 4

Функция четвёртой задачи вычисляет, входит ли число в диапазон -2...2 и выводит результат на экран.





Рисунок 11 — тест программы 4

Остаётся организовать оператор выбора switch. Для этого у пользователя будет спрашиваться номер задания, которое он хочет проверить. Далее switch запускает case, который содержит функцию нужной задачи.

# Приложение

Листинг программы:

#include <iostream>

#include <locale.h>

#include <stdio.h>

#include <cmath>

#define E 2.718281828459045235

#define TI 20

using namespace std;

float kor(float g)

{

int cmp = 1;

int sqr = cmp;

float arif1, arif2, arif3;

while (sqr < g)

{

sqr = cmp;

sqr = cmp \* sqr;

cmp += 1;

}

if (g > 1) cmp--;

if (sqr == g) return cmp;

arif1 = (g / cmp + cmp) / 2;

arif2 = (g / arif1 + arif1) / 2;

arif3 = (g / arif2 + arif2) / 2;

return arif3;

}

float coss(float u)

{

while (u >= 360) u -= 360;

float cos = u;

int bul;

if (u > 90 && u < 270) bul = 0;

else bul = 1;

int i = 0;

if (cos >= 180) cos = 360 - cos;

if (cos >= 90) cos = 180 - cos;

float spis[] = { 1.0,0.984,0.939,0.866,0.766,0.642,0.5,0.342,0.173,0.0 };

for (int ten = 0; cos >= ten; ten += 10) i++;

if ((i - 1) \* 10 == cos) return spis[i - 1];

float ans = spis[i - 1] - ((spis[i - 1] - spis[i]) / 10 \* (cos - 10 \* (i - 1)));

if (bul == 0) ans = 0 - ans;

return ans;

}

float sinn(float u)

{

while (u >= 360) u -= 360;

float cos = u;

int bul;

if (u > 180 && u < 360) bul = 0;

else bul = 1;

int i = 0;

if (cos >= 180) cos = 360 - cos;

if (cos >= 90) cos = 180 - cos;

float spis[] = { 0.0,0.173,0.342,0.5,0.642,0.766,0.866,0.939,0.984,1.0 };

for (int ten = 0; cos >= ten; ten += 10) i++;

if ((i - 1) \* 10 == cos) return spis[i - 1];

float ans = spis[i - 1] + ((spis[i] - spis[i - 1]) / 10 \* (cos - 10 \* (i - 1)));

if (bul == 0) ans = 0 - ans;

return ans;

}

double nlog(double x) {

if (x <= 0) {

printf("Error!");

return 0;

}

int pa = 0;

while (x > 1.0) {

x /= E;

pa++;

}

while (x < .25) {

x \*= E;

pa--;

}

x -= 1.0;

double t = 0.0, s = 1.0, z = x;

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

t += z \* s / k;

z \*= x;

s = -s;

}

return t + pa;

}

void task2()

{

float x, ans, ans1, res = 1, sum = 0;

printf("Введите число x (|x| < 1):\n");

scanf("%f", &x);

if (abs(x) >= 1) {

printf("Неподходящее число!");

return;

}

for (int n = 1; abs(res) > 0.001; n++)

{

ans = pow(-1, n - 1) \* n\* pow(x, n - 1);

if (n > 1) ans1 = pow(-1, n - 2) \* pow(x, n - 2);

if (n > 1) res = ans - ans1;

sum += ans;

}

printf("\nЗначение функции F(%f): %f", x,sum);

}

void task31()

{

int n;

cout << "\n\nПЕРВАЯ ЗАДАЧА\n" << "Введите натуральное число\n";

cin >> n;

cout << "Натуральные делители числа " << n << "\n";

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

{

if (n % i == 0) std::cout << i << " ";

}

printf("\nОшибки:\n Идентефикаторы i и n не определены\n Условие оператора if не заключено в скобки\n Поток не является выражением оператора if\n Отстутсвует использование пространства имён");

printf("\n Условие не является частью цикла for\n Условие цикла задано неправильно\n Инициализаци, условие и изменение оператора for должны отделяться точкой с запятой");

}

void task32()

{

int n;

long long int y = 1;

cout << "\n\nВТОРАЯ ЗАДАЧА\n" << "Введите натуральное число\n";

cin >> n;

if (n % 2 != 0)

{

for (int i = 1; i <= n; i += 2) y \*= i;

}

else for (int i = 2; i <= n; i += 2) y \*= i;

std::cout << "Двойной факториал (n!!) числа " << n << "\n" << y;

printf("\nОшибки:\n Не включена библиотека iostream для потоков\n Вместо вывода из потока переменной n программа вводит её в поток, хотя она ещё не инициализирована\n Инициализировать переменные можно сразу, в той же строке int\n");

printf(" Выражения циклов и условий должны быть заключены в фигурные скобочки {} с двух сторон, без круглых ()\n Для указания условия неравности чисел используется !=, а не <>\n");

printf(" Во всех циклах for не указано изменение\n Отсутствует использование пространства имён\n Использована неициализированная и, в целом, не нужная переменная I\n Переменная i не инициализирована. В циклах нужно добавлять перед ней int\n Условия циклов написаны неправильно\n");

printf(" Двойной факториал вычисляется неправильно, циклы с прибавлением fn к y не имеют смысла, так как fn уже является ответом. fn можно убрать и заменить его на y\n Лучше задать тип переменной y как long long int");

}

void task33()

{

int n;

double p = 1.0;

cout << "\n\nТРЕТЬЯ ЗАДАЧА\n" << "Введите натуральное число\n";

cin >> n;

for (double i = 2; i <= n; i += 2) p = p \* (i / (i + 1));

for (double i = 1; i <= n; i += 2) p = p \* ((i + 1) / i);

std::cout << "Произведение первых " << n << " сомножителей числа " << n << "\n";

std::cout << p;

printf("\nОшибки:\n Операторы for написаны с большой буквы\n Инициализация, условие и изменение оператора for должны отделяться точкой с запятой\n Условия операторов for написаны неверно\n");

printf(" Выражения операторов for не обязательно заключать в фигурные скобки, достаточно написать их после оператора\n Переменные P и p подразумеваются как одна переменная, но написаны с разным размером\n");

printf(" Переменные i, n, p не инициализированы\n Отсутствует использование пространства имён\n Отсутствуют функция main и библиотека iostream\n Вместо p=p\*... можно использовать p\*=...");

}

void task34()

{

int n, x;

double r = 0, res = 0, s = 0;

cout << "\n\nЧЕТВЁРТАЯ ЗАДАЧА\n" << "Введите n\n";

cin >> n;

cout << "Введите x\n";

cin >> x;

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

{

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

{

s = (x + k) / m;

r += s;

}

res += r;

}

cout << "Сумма ряда\n" << res;

printf("\nОшибки:\n Лишнее слово integer и двоеточие перед ним\n Отсутствует использование пространства имён\n Переменная j нигде не используется\n Переменные x,r и res не объявлены\n Условия операторов for написаны неверно\n У операторов for отсутствует изменение\n Отсутствуют функция main() и библиотека iostream\n Переменные m и k должны быть инициализированы в цикле\n Переменная s не объявлена и не инициализирована");

}

void task35()

{

int n;

double sum1 = 0, sum2 = 0;

cout << "\n\nПЯТАЯ ЗАДАЧА\n" << "Введите натуральное число\n";

cin >> n;

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

{

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

{

sum1 += 1.0 / (i + 2 \* j);

}

sum2 += sum1;

}

cout << "Сумма ряда\n" << sum2;

printf("\nОшибки:\n Выражение вложенного цикла делит целое число 1. Стоит записать его с плавающей точкой или указать тип double\n Отсутствует использование пространства имён\n Переменные sum1 и sum2 могут быть инициализированы в первой же строке\n Условие операторов for написаны неверно\n Операторы for должны иметь изменение\n Выражения внутри циклов могут быть записаны как +=\n У циклов for отсутствуют открывающие скобки\n Отсутствуют функция main() и заголовочный файл iostream\n Переменные i и j могут быть инициализированы в цикле вместо глобального объявления");

}

void task36()

{

double b, c, a;

int n = 0;

cout << "\n\nШЕСТАЯ ЗАДАЧА\n" << "a1="; cin >> b;

cout << "a2="; cin >> c;

for (int i = 3; i <= 10; i++)

{

cout << "a" << i << "="; cin >> a;

if (c > b && a < c) n++;

b = c; c = a;

}

cout << "Количество чисел, которые больше, чем их соседи\n" << n;

printf("\nОшибки:\n Отсутствует использование пространства имён\n Отсутствует функция main() и заголовочный файл iostrem\n Переменные c, b, n, a должны быть объявлены\n В инициализации цикла for должен стоят int перед переменной i\n");

printf(" b=c и c=a должны быть вне условия if\n При выводе из потока переменная a зачем-то заключена в скобки\n Вместо запятых в выводе с поток должны использоваться <<\n В выражении цикла используется большая I, хотя в инициализации она маленькая i\n Условие цикла записано неверно\n У цикла отсутствует изменение\n and в условии должен быть стоять внутри скобок между условиями и быть заменён на &&\n n=n++ следует заменить на просто n++\n Переменной i следует присвоить значение 3, так как первые два числа уже были введены");

}

void task37()

{

double p = 1;

cout << "\n\nСЕДЬМАЯ ЗАДАЧА\n" << "i=";

int i; cin >> i;

for (i; i <= 20; i++)

{

for (int j = 1; j <= 20; j++)

{

p += 1.0 / (i + pow(j, 2));

}

}

cout << "Произведение ряда\n" << p;

printf("\nОшибки:\n Функция main должна быть написана с маленькой буквы и иметь тип void\n Отсутствует заголовочный файл iostream и пространство имён\n Переменная i должна задаваться пользователем\n Переменная p должна быть одного размера и быть инициализирована вне цикла\n Отстутсвует заголовочный файл cmath\n Функции sqr() не существует, нужно заменить её на pow(j,2)\n");

printf(" При делении 1 на число следует указать тип 1 как double или float\n Переменной p лушче задать тип double\n Вместо p=p+... лучше использовать p+=...\n Переменную j лучше объявить внутри цикла\n Выражения циклов должны быть заключены в фигурные скобки с двух сторон\n Условия циклов записаны неверно, отсутствует изменение\n for должна быть записана с маленькой буквы\n ");

}

void task4()

{

float x;

printf("Введите число для проверки, входит ли оно в диапазон от -2 до 2:\n");

scanf("%f", &x);

if (x >= -2 && x <= 2) printf("Число входит в диапазон");

else printf("Число не входит в диапазон");

}

void main()

{

system("chcp 1251");

int s;

std::cout << "Введите номер задания, которое хотите проверить (1,2,3,4):\n";

std::cin >> s;

switch (s)

{

case 1:

{

float a, b;

printf("Enter number a:\n");

scanf("%f", &a);

printf("Enter number b:\n");

scanf("%f", &b);

float sqrt = 1 + (E / (a \* a \* a + b \* 3.4));

printf("Значение натурального логарифма от суммы синуса квадрата %f и косинуса %f, делённой на квадратный корень из суммы 1 и числа e, делённого на сумму %f в третьей степени и %f, умноженного на 3,4 равна:\n", a, b, a, b);

printf("ln((sin(a^2)+cos(b))/sqrt(1+e/(a^3+3,4\*b)) = %f", nlog((sinn(a \* a) + coss(b)) / kor(sqrt)));

break;

}

case 2:

task2();

break;

case 3:

{

task31();

task32();

task33();

task34();

task35();

task36();

task37();

break;

}

case 4:

task4();

break;

}

}