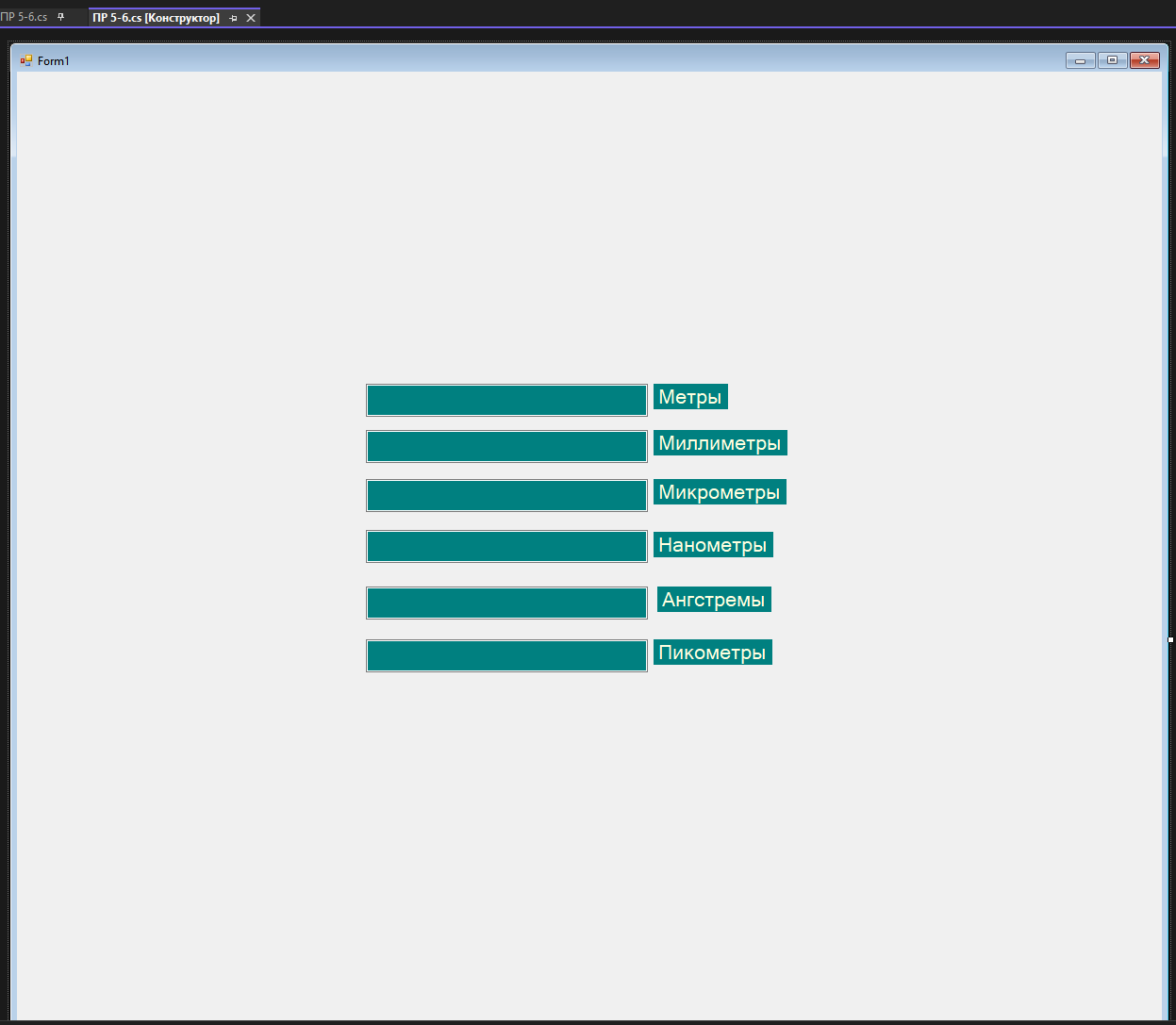
ПРАКТИЧЕСКАЯ РАБОТА №5, 6.

**Отчет:**1.Открыть конструктор, и нажатием клавиш Ctrl + Alt + X открыть панель элементов

2.В Панели элементов найти и открыть 6 TextBox и 6 lable, расставить их в нужном порядке и дать имя всем lable



3. В окне конструктора двойным нажатием мыши на каждый из объектов, чтобы они отобразились в коде

4. Зайти в окно кода, и найти строку

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

5. Вставить в скобки данной строки:

double s;

s = Convert.ToDouble(textBox1.Text);

textBox2.Text = (s \* 1000).ToString();

textBox3.Text = (s \* 1000000).ToString();

textBox4.Text = (s \* 1000000000).ToString();

textBox5.Text = (s \* 10000000000).ToString();

textBox6.Text = (s \* 100000000000).ToString();

1. **private void textBox1\_TextChanged(object sender, EventArgs e)**

**private:** Означает, что этот метод доступен только внутри класса, в котором он объявлен.

**void:** Указывает, что метод не возвращает никакого значения.

**textBox1\_TextChanged:** Это имя метода, которое соответствует событию TextChanged текстового поля textBox1.

**(object sender, EventArgs e):** Параметры, передаваемые методу при возникновении события.

* + - **sender:** Ссылается на объект, который вызвал событие (в данном случае, textBox1).
    - **e:** Содержит аргументы события, которые могут содержать дополнительную информацию о нем.

1. **double s;**

Объявляется переменная s типа double, которая будет использоваться для хранения числового значения, полученного из textBox1.

1. **s = Convert.ToDouble(textBox1.Text);**

**textBox1.Text:** Считывает текст, введенный пользователем в текстовое поле textBox1.

**Convert.ToDouble(...):** Преобразует текст, полученный из textBox1, в число с плавающей точкой типа double.

**s = ...:** Присваивает полученное числовое значение переменной s.

1. **textBox2.Text = (s \* 1000).ToString();**

**s \* 1000:** Умножает число s на 1000.

**.ToString():** Преобразует результат умножения в строку.

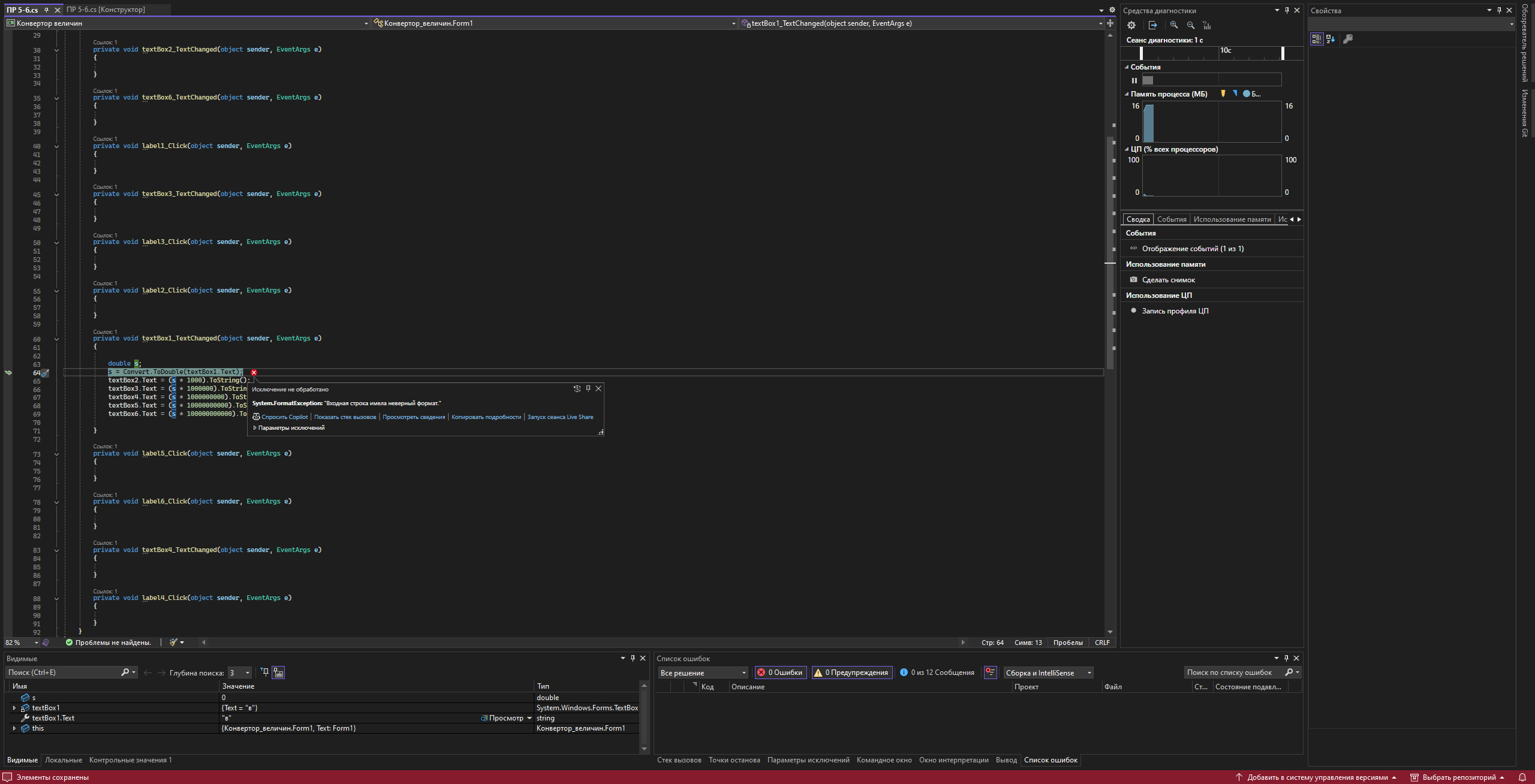
**textBox2.Text = ...:** Присваивает полученную строку свойству Text текстового поля textBox2, отображая результат в этом поле и т.д.

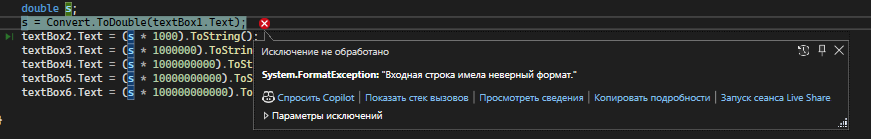
6.Нажать Пуск.

ПРАКТИЧЕСКАЯ РАБОТА №7.

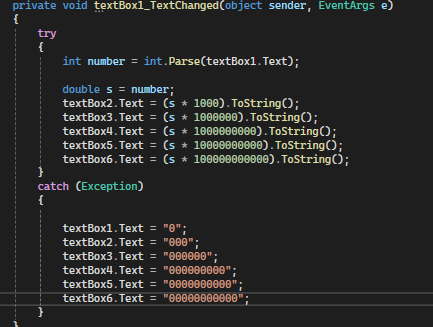
Продолжаем модернизировать практическую №5,6

При вводе значений для расчета конвертора величин, ввели случайно букву, и программа выдаёт ошибку:





Добавить в код:



1.try блок:

**try блок** используется для обработки потенциальных исключений, которые могут возникнуть при выполнении кода внутри него.

**int number = int.Parse(textBox1.Text);**

Эта строка пытается преобразовать текст, введенный в textBox1, в целое число (тип int).

**double s = number;**

Если преобразование успешно, создается переменная **s** типа **double**, которая инициализируется значением **number**.

**textBox2.Text = (s \* 1000).ToString();**

Умножается **s** на 1000, результат преобразуется в строку и присваивается свойству **Text** текстового поля **textBox2** и далее аналогично

**catch (Exception) блок:**

catch блок выполняется, если в try блоке возникло исключение.

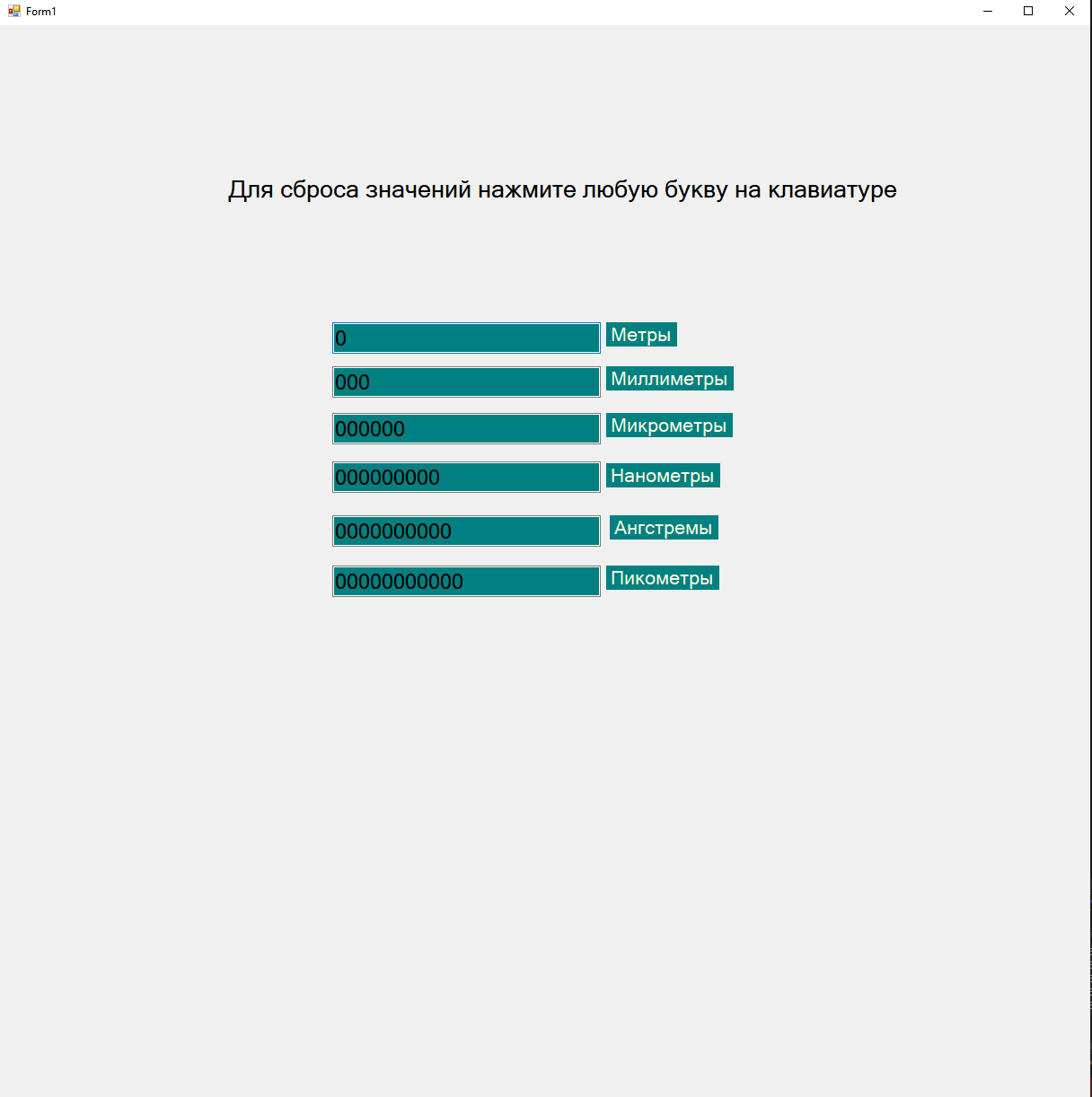
**textBox1.Text = "0";**

Очищается textBox1 и устанавливается в “0”

**textBox2.Text = "000";**

Очищается textBox2 и устанавливается в “000” далее аналогично

При запуске кода и нажатии букв появляются нули:



ПРАКТИЧЕСКАЯ РАБОТА №2.

**Задание 1**: написание кода который будет запрашивать два числа у пользователя и суммировать их

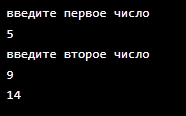
**Код:**

a=int(input("введите первое число"))

b=int(input("введите второе число"))

print(a+b)

**Что он показывает в консоли:**



**Описание:**

**1.**функция int в коде Функция **int** в Python используется для преобразования значений в целые числа. Она может принимать строковые или числовые значения в качестве аргументов.в случае нашего кода он запрашивает число которое прикрепится к значению а после чего запросит и значение b

2. функция print выводит на экран заявленную операцию или текст, в нашем случае он вывел на экран сумму a b при помощи оператора +

**Задание 2: написать программу которая будет определять является ли число простым**

**Код:**

**a = int(input("Введите число: "))**

**k = 0**

**for i in range(2, a // 2+1):**

**if (a % i == 0):**

**k = k+1**

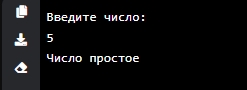
**if (k <= 0):**

**print("Число простое")**

**else:**

**print("Число не является простым")**

**что он показывает в консоли:**



**Описание:**

Пользователь вводит число, и оно сохраняется в переменную a.

Инициализируем переменную k значением 0. Эта переменная будет выполнять роль счетчика.

Запускаем цикл for в диапазоне от 2 до значения проверяемого числа, деленного на 2 (речь идет, конечно, о целочисленном делении). Напоминаем, что само число и 1 делителями мы считать не будем.

Затем, при помощи инструкции if, на каждой итерации цикла мы проверяем, делится ли наше число без остатка на числа из выбранного диапазона цикла. Если делится, то переменная k, выполняющая роль счетчика, увеличивается на единицу.

Если число делителей равно 0, то проверяемое число является простым.

Выводим полученный результат на экран.

**Задание 4: написать код который будет находить корни квадратного уравнения**

**Код:**

**import math**

**print("Введите коэффициенты для уравнения")**

**print("ax^2 + bx + c = 0:")**

**a = float(input("a = "))**

**b = float(input("b = "))**

**c = float(input("c = "))**

**discr = b \*\* 2 - 4 \* a \* c**

**print("Дискриминант D = %.2f" % discr)**

**if discr > 0:**

**x1 = (-b + math.sqrt(discr)) / (2 \* a)**

**x2 = (-b - math.sqrt(discr)) / (2 \* a)**

**print("x1 = %.2f \nx2 = %.2f" % (x1, x2))**

**elif discr == 0:**

**x = -b / (2 \* a)**

**print("x = %.2f" % x)**

**else:**

**print("Корней нет")**

**описание кода:**

Импорт модуля math эта строка импортирует модуль math, который предоставляет функции для математических операций, таких как sqrt (извлечение квадратного корня).

Ввод коэффициентов Программа просит пользователя ввести значения коэффициентов a, b и c для квадратного уравнения.

3.Вычисление дискриминанта Код вычисляет дискриминант (D) по формуле b² - 4ac. Дискриминант определяет количество решений квадратного уравнения:

Если D > 0, уравнение имеет два различных корня.

Если D = 0, уравнение имеет один корень (кратный).

Если D < 0, уравнение не имеет действительных корней.

4. поиск решений

Программа проверяет дискриминант и вычисляет корни уравнения по формулам:

Если D > 0, вычисляются два корня x1 и x2.

Если D = 0, вычисляется единственный корень x.

Если D < 0, выводится сообщение, что корней нет.

5. вывод результатов

В зависимости от значения дискриминанта программа выводит найденные корни или сообщение о том, что корней нет.

Задание 5

Задача: написать программу которая генерирует случайное число в заданном диапозоне

Код: import random

def generate\_random\_number(start, end):

return random.randint(start, end)

# Задайте диапазон

start\_range = int(input("Введите начало диапазона: "))

end\_range = int(input("Введите конец диапазона: "))

random\_number = generate\_random\_number(start\_range, end\_range)

print(f"Случайное число в диапазоне от {start\_range} до {end\_range}: {random\_number}")

описание:

Строка импортирует модуль рандом, который предоставляет функции для работы со случайными числами

здесь мы определяем функцию generate\_random\_number, которая принимает два аргумента: start и end. Эти аргументы представляют собой границы диапазона, в котором мы хотим сгенерировать случайное число.

Внутри функции мы используем random.randint(start, end), чтобы сгенерировать и вернуть случайное целое число, которое находится в заданном диапазоне, включая сами границы (то есть start и end).

Здесь мы запрашиваем у пользователя ввод значений для начала и конца диапазона. Функция input() считывает ввод пользователя в виде строки, а функция int() преобразует эту строку в целое число. Эти значения сохраняются в переменные start\_range и end\_range.

В этой строке мы вызываем нашу функцию generate\_random\_number, передавая ей ранее введенные значения start\_range и end\_range. Результат (случайное число) сохраняется в переменной random\_number.

Наконец, мы выводим результат на экран, используя f-строку для форматирования. Эта строка вставляет значения переменных start\_range, end\_range и random\_number в текст, который будет выведен.

Результаты:

