1. Реализовать функцию, принимающую два числа (позиционные аргументы) и выполняющую их

деление. Числа запрашивать у пользователя, предусмотреть обработку ситуации деления на

ноль.

2. Выполнить функцию, которая принимает несколько параметров, описывающих данные

пользователя: имя, фамилия, год рождения, город проживания, email, телефон. Функция

должна принимать параметры как именованные аргументы. Осуществить вывод данных о

пользователе одной строкой.

3. Реализовать функцию **my\_func()**, которая принимает три позиционных аргумента и

возвращает сумму наибольших двух аргументов.

4. Программа принимает действительное положительное число **x** и целое отрицательное число

**y**. Выполните возведение числа **x** в степень **y**. Задание реализуйте в виде функции

**my\_func(x, y)**. При решении задания нужно обойтись без встроенной функции возведения

числа в степень.

Подсказка: попробуйте решить задачу двумя способами. Первый — возведение в степень с

помощью оператора \*\*. Второй — более сложная реализация без оператора \*\*,

предусматривающая использование цикла.

5. Программа запрашивает у пользователя строку чисел, разделённых пробелом. При нажатии

Enter должна выводиться сумма чисел. Пользователь может продолжить ввод чисел,

разделённых пробелом и снова нажать Enter. Сумма вновь введённых чисел будет

добавляться к уже подсчитанной сумме.

Но если вместо числа вводится специальный символ, выполнение программы завершается.

Если специальный символ введён после нескольких чисел, то вначале нужно добавить сумму

этих чисел к полученной ранее сумме и после этого завершить программу.

6. Реализовать функцию **int\_func()**, принимающую слова из маленьких латинских букв и

возвращающую их же, но с прописной первой буквой. Например, **print(int\_func(‘text’))** -> Text.

7. Продолжить работу над заданием. В программу должна попадать строка из слов, разделённых

пробелом. Каждое слово состоит из латинских букв в нижнем регистре. Нужно сделать вывод

исходной строки, но каждое слово должно начинаться с заглавной буквы. Используйте

написанную ранее функцию **int\_func()**.

GPT

**Задача: Мини-калькулятор со встроенными проверками**

Напиши программу, которая:

1. Запрашивает у пользователя строку вида:
2. число операция число

Например:

15 + 7

10 / 0

3 \*\* 4

1. Поддерживает операции: +, -, \*, /, \*\* (возведение в степень).
2. Использует **функции** для:
   * парсинга строки;
   * выполнения операции;
   * запуска основного цикла работы калькулятора.
3. Обрабатывает ошибки через try / except:
   * деление на ноль;
   * неверный ввод (например, a + b);
   * неизвестная операция.
4. Работает в цикле до тех пор, пока пользователь не введет exit.

Теперь калькулятор должен уметь работать не только с двумя числами, а с выражением **любой длины**.

📌 Пример:

Ввод: 2 + 3 \* 4 - 5 / 2

Вывод: 11.5

### Условия:

1. Пользователь вводит строку выражения (2 + 3 \* 4 - 5 / 2).
2. Нужно правильно обработать приоритет операций (\* и / выше, чем + и -, а \*\* ещё выше).
3. Ошибки по-прежнему обрабатываются (деление на 0, неправильный ввод, лишние символы).
4. Работает в цикле, выход по exit.
5. **Нельзя использовать eval() и exec()** — нужно написать логику самому.