**Практическая работа №7. Функции. Область видимости переменных**

**Выполнил студент группы ИСП23.1А Сологуб Денис**

**Задача 1**

**Формат ввода**

Вводится строка слов, записанных через +, окруженный пробелами, затем вводится число.

**Формат вывода**

Через запятую и пробел большими буквами выведите слова с длиной, кратной числу. Выводить в том же порядке, что и в строке.

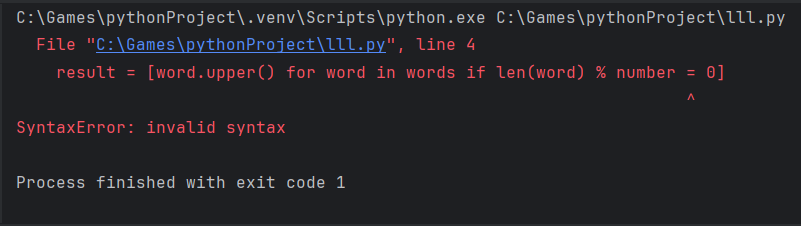
|  |  |  |  |
| --- | --- | --- | --- |
| Имя переменной | Тип | Назначение | Уровень видимости |
| Words | List | Список слов, полученный из строки ввода | Локальная |
| Number | Int | Число, по которому проверяется кратность длинны слов | Локальная |
| result | list | Список слов, длина которых кратна number, в верхнем регистре | Локальная |

**Словесный алгоритм решения:**

1. Ввод данных — Считать строку слов, разделённых +, из стандартного ввода. Считать число, по которому будет проверяться длина слов.
2. Обработка данных — Разделить строку на отдельные слов. Преобразовать введённое число в целочисленный формат. Отфильтровать слова, длина которых кратна введённому числу. Преобразовать отобранные слова в верхний регистр.
3. Вывод результата — Если найденные слова есть, вывести их, разделяя занятой и пробелом. Если таких слов нет, вывести сообщение.

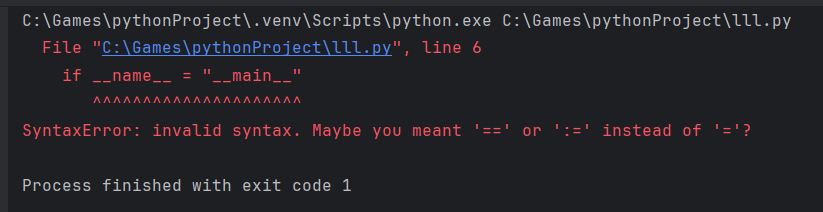
**Код:**

def main():  
 words = input().strip().split(' + ')  
 number = int(input().strip())  
 result = [word.upper() for word in words if len(word) % number == 0]  
 print(', '.join(result) if result else "Нет подходящих слов")  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Ошибки:** 

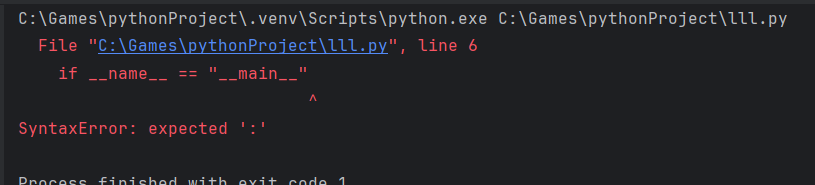
Ошибка: не хватает еще одного знака «=». Из-за чего код присваивал значение 0, когда нужно было сравнить.

Исправление: добавить перед 0 еще один знак «=»



Ошибка: не хватает еще одного знака «=».

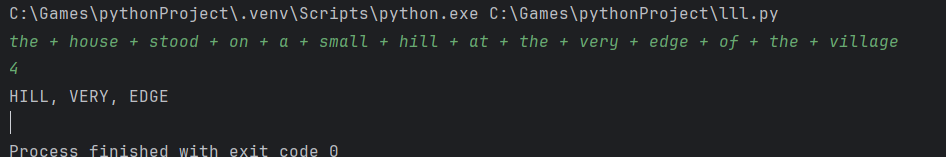
Исправление: Добавление перед main знак «=».

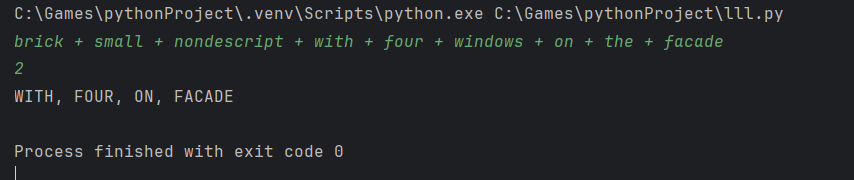
****

Ошибка: не хватает знака «:» после main. Из-за этого программа не смогла понять, что я хотел сделать блок кода с отступом.

Исправление: Добавление в код знак «:».

**Тесты:**

****

****

**Задача 2**

**Формат ввода**

Вводится три строки слов, записанных через пробел.

**Формат вывода**

В произвольном порядке без повторений через запятую и пробел вывести слова, которые были во всех строках.

Затем вывести слово, позже всех идущее по алфавиту среди подходящих.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя переменной | Тип | Назначение | Уровень видимости |
| Words1 | Set | Множество слов из первой строки | Локальная |
| Words2 | Set | Множество слов из второй строки | Локальная |
| Words3 | Set | Множество слов из третьей строки | Локальная |
| Common\_words | set | Множество слов, которые есть во всех трёх строках | Локальная |

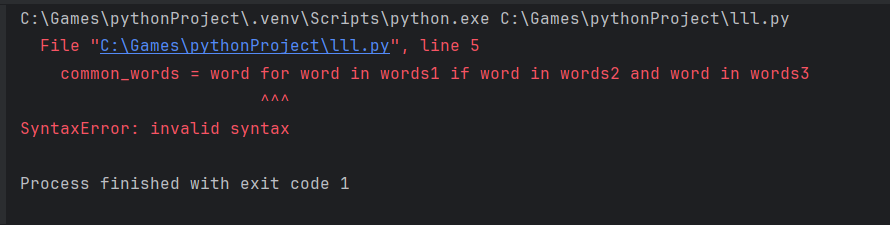
**Словесный алгоритм решения:**

1. Считать входные данные — Прочитать три строки, содержащие слова, разделённые пробелами. Преобразовать каждую строку в множество, чтобы удалить повторяющиеся слова
2. Найти общие слова — Для каждого слова из первой строки проверить, есть ли оно во второй и третьей.Сохранить такие слова в отдельное множество.
3. Вывести результат — Если есть общие слова, вывести их в алфавитном порядке через запятую. Вывести слово, которое идет последним по алфавиту среди найденных. Если общих слов нет, вывести сообщение.

**Код:**

def main():  
 words1 = set(input().split())  
 words2 = set(input().split())  
 words3 = set(input().split())  
 common\_words = {word for word in words1 if word in words2 and word in words3}  
 print(", ".join(sorted(common\_words)) if common\_words else '')  
 print(max(common\_words) if common\_words else '')  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

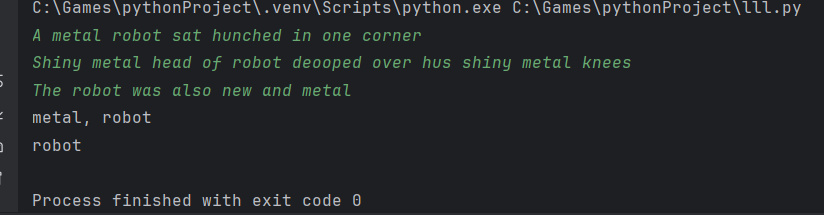
**Ошибки:**

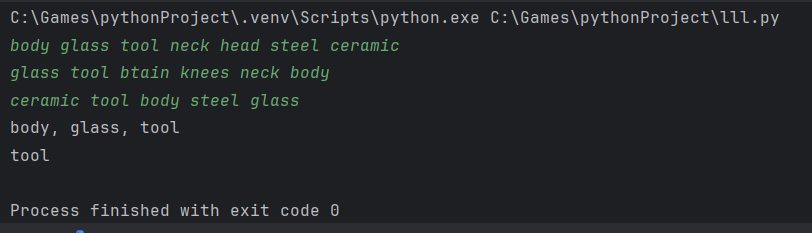


Ошибка: не поставлены {} для множеств set, чтобы убирать дубликаты и фильтровать слова.

Исправление: Добавить символы «{}»

**Тесты:**

****

****