

Задачи на повторение.

1

Счастливый билет 🚌

Серийный номер билета на автобус - это шестизначное число, например 432190. Счастливым билетом считается билет, у которого сумма первых трех цифр серийного номера равно сумме последних трех цифр.

- Напишите функцию **is_lucky_ticket()**, которая принимает один аргумент `ticket_id (int)` - серийный номер билета.
- Функция должна возвращать `True`, если билет является счастливым, и `False` в противном случае.

Дополнительно: Напишите подробную строку документации для этой функции, а также сделайте проверку на то, что `ticket_id` является целым числом (`TypeError`) и это число шестизначное (`ValueError`). Если это не так, то программа должна вбросить соответствующее исключение. p.s. Так можно делать и для всех остальных функций.

2

Числа Армстронга 🧐

Число Армстронга — это число, сумма кубов цифр которого равна самому числу (например, $153 = 1^3 + 5^3 + 3^3$).

- Напишите программу, которая вычисляет все числа Армстронга в диапазоне от 100 до 1000 и выводит их на экран в виде списка (их должно быть 4) .

3

Цензор ✖

- Напишите функцию **sensor_print()**, которая принимает один аргумент - **string** (`str`).
- Функция печатает `string` в консоль, но при этом заменяет “запрещенные” слова в этой строке на “***”. (Список с запрещенными словами должен находиться внутри функции, составьте его сами).

Например, если список запрещенных слов выглядит так: ["запрещенное", "слово"], то такой вызов `sensor_print("Это запрещенное слово и еще одно запрещенное слово.")` должен вывести в консоль "Это *** *** и еще одно *** ***."

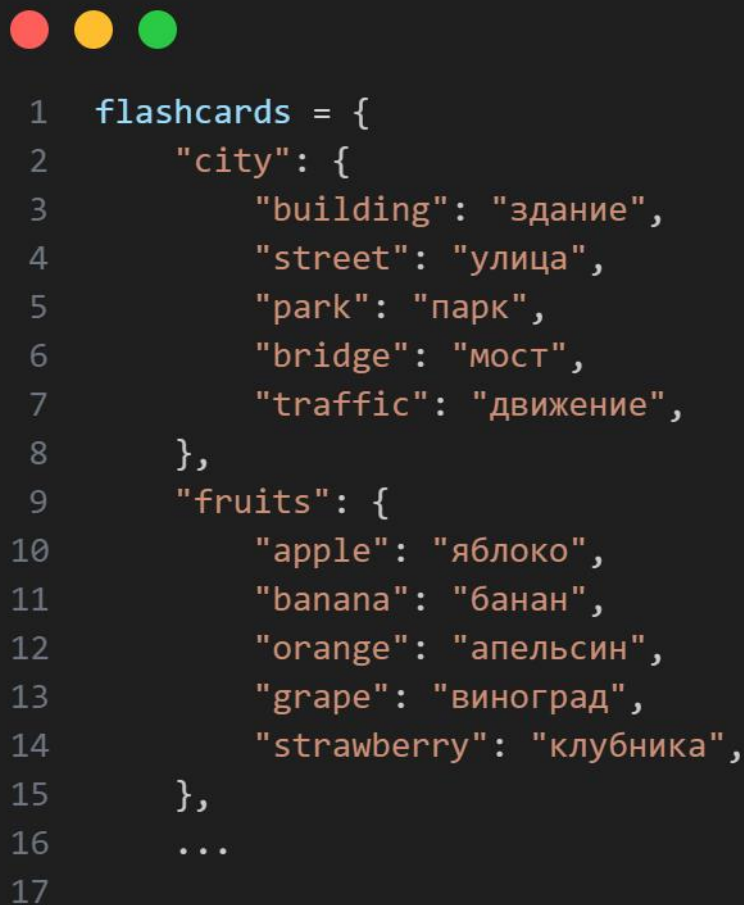
p.s. Для этой задачи удобно использовать один из методов строк.

4*

D...  

- Создайте словарь **flashcards**, где ключи являются "темами", а значения - это словари, где ключом является слово на английском языке, а значение - его перевод на русский или любой другой язык.

Например:



```
1  flashcards = {
2      "city": {
3          "building": "здание",
4          "street": "улица",
5          "park": "парк",
6          "bridge": "мост",
7          "traffic": "движение",
8      },
9      "fruits": {
10         "apple": "яблоко",
11         "banana": "банан",
12         "orange": "апельсин",
13         "grape": "виноград",
14         "strawberry": "клубника",
15     },
16     ...
17
```

- Программа должна (в бесконечном цикле) приветствовать пользователя и предложить ввести одну из существующих тем, например “Привет! Выбери сегодняшнюю тему для изучения: city, fruits... или введи **выйти**”
- Если пользователь ввел “выйти”, то программа выводит на экран “Но ты приходи еще!”, выходит из бесконечного цикла и завершается.
- Если пользователь ввел что-то, кроме предложенных тем или слова “выйти”, то программа должна просто попросить его ввести строку еще раз.
- Если пользователь ввел одну из предложенных тем, то начинается “игра”. Программа создает счетчик правильных ответов и начинает последовательно просить написать перевод каждого слова из выбранной тематики, например “Как на русском будет **apple**?”.

- Если пользователь вводит правильный перевод, то программа пишет “Верно!”, увеличивает счетчик правильных ответов на 1 и задает следующий вопрос.
- Если пользователь вводит неверный ответ, то программа пишет f“Неверно! Правильный перевод: {правильный перевод}”.
- После того, как закончились все вопросы, программа выводит на экран количество правильных ответов и количество вопросов, например f“Ты набрал {количество_верных_ответов} из {количество_вопросов}.”. После этого программа возвращается к первоначальному вопросу про выбор тематики.

Дополнительно, по желанию:

- Можно держать вопросы в отдельном модуле и импортировать в главную программу. Или создать файл **json** и загружать вопросы из него. Так программа будет выглядеть более структурированной и читаемой.
- Можно расширить функционал, например сохранять неправильные ответы в список и в конце игры выводить их на экран и тд.