

Лабораторная работа №2

1. Напиши функцию `process_words(words)`, которая:

- приводит все слова к нижнему регистру (`map + lambda`),
- оставляет только слова длиной > 3 (`filter + lambda`),
- сортирует по алфавиту.

Пример использования функции:

```
words = ["Python", "is", "Great", "AI"]
print(process_words(words))
# ['great', 'python']
```

2. Сделать функцию `calc(expr)`, которая принимает строку с выражением вида `"2 + 3 * 4 - 5"`.

- Разбивает строку,
- Использует словарь лямбд для операций (`+`, `-`, `*`, `/`),
- Последовательно считает результат.

Дополнительно: поддержка скобок.

3. Функция `make_password_generator(length)` возвращает вложенную функцию, которая при каждом вызове генерирует новый пароль из случайных букв/цифр.

```
gen = make_password_generator(8)
print(gen()) # e.g. "a9X2kLmP"
```

4. Написать программу, которая принимает текст и:

- считает количество слов и символов,
- строит словарь частот слов,
- выводит топ-N самых частых слов,
- умеет фильтровать слова по условию (например, длина больше 3).

5. Есть список событий:

```
events = [  
    {"title": "Math", "start": "09:00", "end": "10:30"},  
    {"title": "English", "start": "10:00", "end": "11:00"}  
]
```

Требования:

- Проверить корректность времени (`start < end`).
- Найти пересекающиеся события.
- Отсортировать по времени начала.

Ожидаемый вывод:

Конфликт: Math (09:00–10:30) и English (10:00–11:00)

6. Симуляция игры в «Кости»:

Цель: симуляция серии партий с балансом и ставками.

Вход: стартовый баланс, число раундов, стратегия ставки.

Выход: финальный баланс, лог по раундам.

Правила:

- Игрок и компьютер бросают по 2 кубика $\rightarrow \text{sum in } [2..12]$.
- Побеждает тот, у кого сумма больше; при равенстве — ничья.
- Перед раундом игрок делает ставку `bet` (валидация $0 < \text{bet} \leq \text{balance}$).
- Выигрыш: $+\text{bet}$, проигрыш: $-\text{bet}$, ничья: 0.

Функции:

- `roll()` \rightarrow случайная сумма 2..12.
- `play_round(bet)` \rightarrow результат раунда и изменение баланса.
- `run_game(rounds, bet_strategy)` \rightarrow цикл; `bet_strategy(balance, round_idx)` \rightarrow `bet`.

Краевые случаи: недостаточно средств; стратегия с нулевой/негативной ставкой — отклонять.

Вывод (пример):

```
R1: you=7, cpu=9  $\rightarrow$  -100, balance=900  
R2: you=11, cpu=4  $\rightarrow$  +100, balance=1000  
...  
Final balance: 1150
```