Практические занятие №3-4

1. Форматирование кода

```
In [ ]: # установка библиотеки для проверки кода на соответствие PEP8
        !pip install flake8 pycodestyle_magic
```

```
In [ ]: # теперь код будет проверяться на соответствие PEP8
        %load_ext pycodestyle_magic
        %pycodestyle_on
```

1.1. (0 баллов)

Приведите примеры кода, которые соответствуют нарушениям PEP 8. Для самостоятельной проверки возможно использовать библиотеки pycodestyle для разных сред или pycodestyle_magic для интерактивной среды Jupyter Notebook.

1. `E211 whitespace before '('`
2. `E225 missing whitespace around operator`
3. `E231 missing whitespace after ','`
4. `E251 unexpected spaces around keyword / parameter equals`
5. `E302 expected 2 blank lines, found 1`
6. `E701 multiple statements on one line (colon)`
7. `E702 multiple statements on one line (semicolon)`
8. `E711 comparison to None should be 'if cond is None:'`
9. `E712 comparison to True should be 'if cond is True:' or 'if cond:'`

2. Некоторые особенности языка

2.1. (0 баллов)

Объясните, что это за код. Это Питон?

`0xfor _ in 'abc'`

2.2. (0 баллов)

Странные результаты приведены ниже. Как такое возможно?

```
In [1]: a = 1
        b = 1
        c = 300000 # проверено в Python v3.11
        d = 300000
        print(a is b, c is d)

        True False
```

```
In [ ]: a, b = 'py', 'py'
        c = ''.join(['p', 'y'])
        print(a is b, a == c, a is c)

        True True False
```

2.3. (0.1 балла)

В следующем фрагменте по индексу i выбирается одна из трех строк. Сократите код во второй строке **ровно** до 19 символов без использования функций и методов.

```
In [12]: i = 0
         ['much','code','wow'][i] # 24 символа

Out[12]: 'much'
```

3. Однострочники

Напишите программы-однострочники для следующих задач. Начните с наивных решений и постепенно двигайтесь в сторону решений кратких и изящных. Определитесь, в каких случаях решение-однострочник является предпочтительным, а в каких – нет.

3.1. (0.1 балла)

Преобразовать элементы списка s из строковой в числовую форму.

3.2. (0.1 балла)

Подсчитать количество различных элементов в последовательности s.

3.3. (0.1 балла)

Обратить последовательность s без использования функций.

3.4. (0.1 балла)

Выдать список индексов, на которых найден элемент x в последовательности s.

3.5. (0.1 балла)

Сложить элементы списка s с четными индексами.

3.6. (0.1 балла)

Найти строку максимальной длины в списке строк s.

3.7. (0.1 балла)

Проверить, относится ли число к числам харшад (делящиеся нацело на сумму своих цифр).

3.8. (0.2 балла)

Реализовать функцию-однострочник для RLE-сжатия. Разрешается использование модуля itertools. Пример работы:

```
>>> rle_encode('ABBCCCDEF')
[('A', 1), ('B', 2), ('C', 3), ('D', 1), ('E', 1), ('F', 1)]
```

4. Снова математика...

4.1. (0.2 балла)

Реализовать функцию, вычисляющую поэлементное произведение матриц (произведение Адамара):

$$\mathrm{multiply}(\mathbf{A}, \mathbf{B}) = \mathbf{A} \odot \mathbf{B}, \tag{1}$$

вычислить значение функции при:

$$\mathbf{A} = \begin{pmatrix} 0 & 2 \\ 3 & 0 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 1 & 4 \\ 2 & 0 \end{pmatrix}.$$

Внешний тест:

- $multiply(\mathbf{A}, \mathbf{B}) = [[0, 8], [6, 0]]$

4.2. (0.2 балла)

Реализовать функцию, транспонирующую матрицу:

$$\mathrm{transpose}(\mathbf{A}) = \mathbf{A}^T, \tag{2}$$

вычислить значение функции при:

$$\mathbf{A} = \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 3 \\ 0 & 1 & 1 \end{pmatrix}.$$

Элемент матрицы $\mathbf{C} = \mathrm{transpose}(\mathbf{A})$ с индексами ij вычисляют согласно: $c_{ij} = a_{ji}$.

Внешний тест:

- $transpose(\mathbf{A}) = [[0, 1, 0], [2, 0, 1], [1, 3, 1]]$

4.3. (0.3 балла)

Реализовать функцию, вычисляющую произведение матриц:

$$\mathrm{dot}(\mathbf{A}, \mathbf{B}) = \mathbf{AB}, \tag{3}$$

вычислить значение функции при:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}.$$

Элемент матрицы $\mathbf{C} = \mathrm{dot}(\mathbf{A}, \mathbf{B})$ с индексами ij вычисляют согласно:

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}.$$

Внешний тест:

- $dot(\mathbf{A}, \mathbf{B}) = [[9, 12, 15], [19, 26, 33], [29, 40, 51]]$

5. Ассорти из небольших задач

5.1. (0.2 балла)

Напишите функцию generate_groups(), которая компактно генерирует (а не просто выдает готовый) список всех названий групп в том виде, который используется на сайте ЦАП.

```
['ИВБО-01-22', 'ИВБО-02-22', 'ИВБО-03-22', 'ИВБО-04-22', 'ИВБО-05-22', 'ИВБО-06-22', 'ИВБО-07-22', 'ИВБО-08-22', 'ИКБО-01-22',
'ИКБО-02-22', ...]
```

5.2. (0.2 балла)

Реализуйте свою версию print(). Постарайтесь использовать максимум возможностей настоящей print(). Для вывода используйте функцию sys.stdout.write().

5.3. (0.4 балла)

Вы получили зашифрованное сообщение и теперь предстоит его расшифровать:

```
E3238557 6204A1F8 E6537611 174E5747
5D954DA8 8C2DFE97 2911CB4C 2CB7C66B
E7F185A0 C7E3FA40 42419867 374044DF
2519F07D 5A0C24D4 F4A960C5 31159418
F2768EC7 AEAF14CF 071B2C95 C9F22699
FFB06F41 2AC90051 A53F035D 830601A7
EB475702 183BAA6F 12626744 9B75A72F
8DBFBFEC 73C1A46E FFB06F41 2AC90051
97C5E4E9 B1C26A21 DD4A3463 6B71162F
8C075668 7975D565 6D95A700 7272E637
```

Известно, что для зашифрования использовался алгоритм TEA. Известен также ключ зашифрования/расшифрования:

```
k = [0, 4, 5, 1]
```

Имеется и функция на C/C++ для расшифровки данных (v – слова данных, k – ключ):

```
void decrypt(uint32_t v[2], const uint32_t k[4]) {
    uint32_t v0 = v[0], v1 = v[1], sum = 0xC6EF3720, i;
    uint32_t delta = 0x9E3779B9;
    uint32_t k0 = k[0], k1 = k[1], k2 = k[2], k3 = k[3];
    for (i = 0; i < 32; i++) {
        v1 -= ((v0 << 4) + k2) ^ (v0 + sum) ^ ((v0 >> 5) + k3);
        v0 -= ((v1 << 4) + k0) ^ (v1 + sum) ^ ((v1 >> 5) + k1);
        sum -= delta;
    }
    v[0] = v0; v[1] = v1;
}
```

Остается перевести функцию decrypt на Питон и узнать, что же содержится в зашифрованном сообщении!

6. Генераторы текстов

6.1. (0.3 балла)

Реализуйте генератор докладов по цифровой экономике. Данные для генерации извлекаются из таблицы:

1	2	3	4	5
Коллеги,	парадигма цифровой экономики	открывает новые возможности для	дальнейшего углубления	знаний и компетенций.
В то же время,	контекст цифровой трансформации	выдвигает новые требования	бюджетного финансирования	непроверенных гипотез.
Однако,	диджитализация бизнес-процессов	несёт в себе риски	синергетического эффекта	волатильных активов.
Тем не менее,	прагматичный подход к цифровым платформам	расширяет горизонты	компрометации конфиденциальных	опасных экспериментов.
Следовательно,	совокупность сквозных технологий	заставляет искать варианты	универсальной коммодитизации	государственно-частных партнёрств.
Соответственно,	программа прорывных исследований	не оставляет шанса для	несанкционированной кастомизации	цифровых следов граждан.
Вместе с тем,	ускорение блокчейн-транзакций	повышает вероятность	нормативного регулирования	нежелательных последствий.
С другой стороны,	экспоненциальный рост Big Data	обостряет проблему	практического применения	внезапных открытий.

Придумайте способ автоматизировать извлечение данных таблицы из этого документа.

Доклад должен иметь корректное начало и несколько абзацев текста.

Пример доклада показан ниже:

```
Коллеги, совокупность сквозных технологий повышает вероятность нормативного регулирования опасных экспериментов. С другой
стороны, совокупность сквозных технологий несёт в себе риски нормативного регулирования внезапных открытий. Тем не менее,
совокупность сквозных технологий открывает новые возможности несанкционированной кастомизации опасных экспериментов.

Тем не менее, парадигма цифровой экономики открывает новые возможности практического применения цифровых следов граждан.
Соответственно, совокупность сквозных технологий открывает новые возможности универсальной коммодитизации волатильных
активов. Следовательно, программа прорывных исследований несёт в себе риски практического применения государственно-частных
партнёрств.

Следовательно, ускорение блокчейн-транзакций расширяет горизонты универсальной коммодитизации знаний и компетенций. Вместе
с тем, прагматичный подход к цифровым платформам не оставляет шанса для нормативного регулирования непроверенных гипотез.
Следовательно, прагматичный подход к цифровым платформам повышает вероятность бюджетного финансирования волатильных
активов.
```

6.2. (0.3 балла)

Реализуйте генератор случайных данных ФИО. Список распространенных имен можно скачать из интернета. Фамилии необходимо генерировать самостоятельно – так, чтобы результат оказался не менее реалистичным, чем в примере ниже. Впрочем, можете попробовать генерировать и имена.

Примеры работы генератора:

```
Данил А. Фуций
Роман Д. Фецачли
Лев Ц. Шолодяк
Ильдар Ц. Тачасяк
Николай Р. Мугодич
Данила И. Табян
Семен Л. Черев
Самир Д. Тифомский
Тамерлан О. Гузянц
Святослав И. Набян
Герман З. Семидяк
Тамерлан К. Башко
Назар Ш. Кунарий
Степан Е. Гидский
Леонид Н. Сабин
```

7. Движок для текстовых игр

7.1. (0.4 балла)

Мир простейших текстовых игр (книг-игр, Choose Your Own Adventure) состоит из множества комнат.

Каждая комната содержит:

- название,
- метку для перехода в эту комнату,
- описание комнаты,
- список действий со стороны игрока, приводящих к выводу сообщения и переходу в другую комнату по ее метке.

В рассматриваемом движке для описания игрового мира используется подмножество формата Markdown.

Вот простейшая игра, реализованная таким образом.

Задача состоит в реализации игры подобного типа на основе языка Python. Подразумевается реализация не менее 10 комнат с заданными комнатами начала и конца игры. При этом вариант "победы" не обязательно предусматривается в конечной комнате.

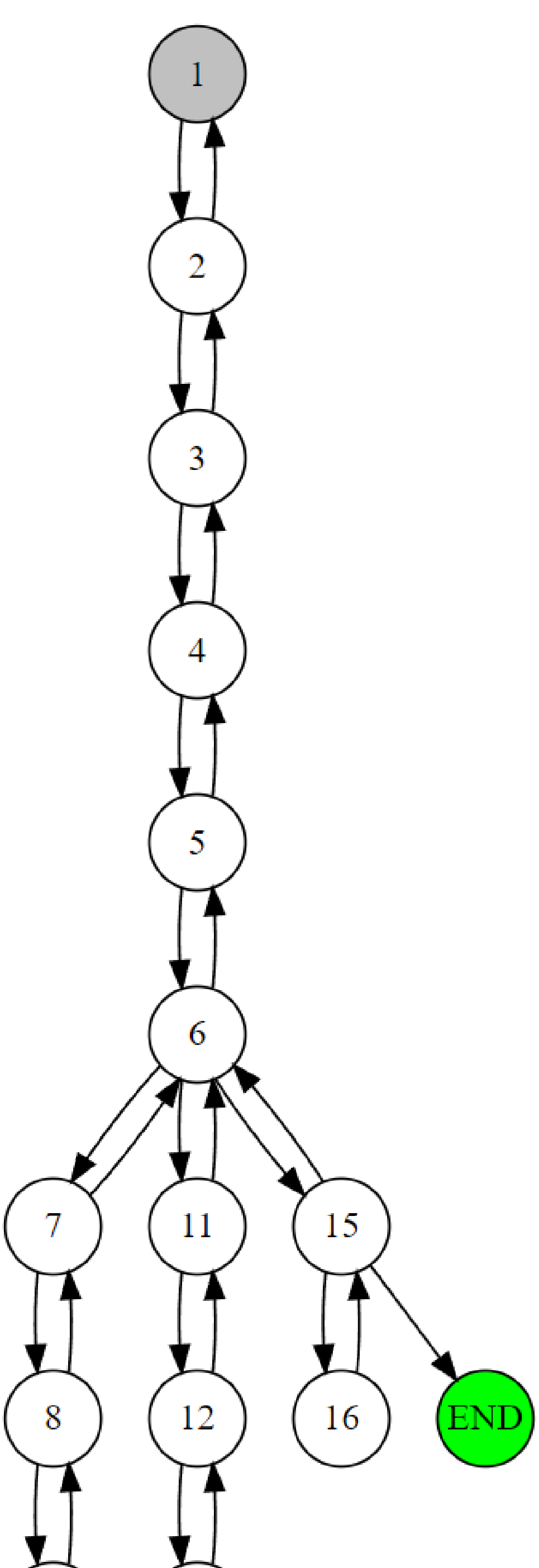
Ниже показан пример сеанса игры:

```
Начало лабиринта

Вы в начале лабиринта. Сможете ли из него выбраться?

1. Проход на запад.

> 1

...

Комната №2

Вы находитесь в комнате №2.

1. Проход на запад.
2. Проход на восток.

> 1

...

Комната №3

Вы находитесь в комнате №3.

1. Проход на север.
2. Проход на восток.

>
```

7.2. (0.5 балла)

Изобразите граф игрового мира из задачи 7.1. с помощью генерации кода на языке Dot – представления инструмента Graphviz. Также можно воспользоваться онлайн версией Graphviz.

Пример вывода графа для game.md:



7.3. (0.5 балла)

Реализуйте инструмент для автоматической проверки игрового мира из задачи 7.1. на наличие тупиков – мест, из которых нельзя добраться до завершения игры.

Распределение сложных задач:

Задача 1 (если не успеваем на занятии): 5.1. + 5.2. + 5.3. - итого 0.8 балла.

Задача 2: 6.1. + 6.2. - итого 0.6 балла.

Задача 3: 7.1. + 7.2. + 7.3. - итого 1.4 балла.