

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

Отчет о лабораторной работе № 3.1 Работа с IPython и Jupyter Notebook

Выполнил:

Шальнев Владимир Сергеевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры
прикладной математики и
компьютерной безопасности,
Воронкин Р.А.

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ:

Счастливый билетик



Билет считается счастливым, если выполнено следующее условие: сумма первых трёх цифр номера равна сумме последних трёх цифр.

Задание:

- 1) Определите число `ticket_number` — шестизначный номер билета;
- 2) Напишите код, который по шестизначному номеру `ticket_number` билета проверяет, является ли он счастливым;
- 3) Если номер счастливый, выведите строку `Yes`, иначе — `No`.

Первая задача

Пример 1:

Input: 123456

Output: No

Пример 2:

Input: 123042

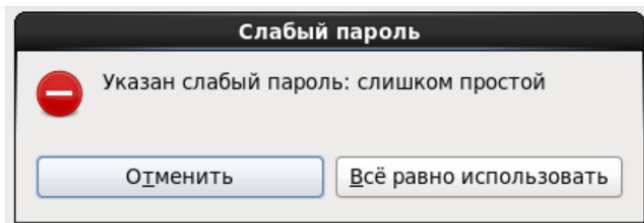
Output: Yes

Ввод [21]: `ticket_number = 123042`

```
Ввод [22]: ticket_number = list(map(int, str(ticket_number)))
            if sum(ticket_number[0:3]) == sum(ticket_number[3:]):
                print('Yes')
            else:
                print('No')
```

Yes

Решение первой задачи



Пусть пароль может содержать только латинские буквы, знаки препинания и цифры.

Пароль считается надёжным, если удовлетворяет следующим условиям:

- содержит буквы в разных регистрах;
- содержит цифры;
- содержит не менее 4 уникальных символов;
- не содержит ваше имя латиницей, записанное буквами любых регистров (anna, iVan, ...).

Иначе пароль считается слабым.

Задание:

- 1) Определите строку `password` — придуманный вами пароль;
- 2) Напишите код, который по паролю `password` проверяет, является ли он надёжным;
- 3) Если пароль надёжный, выведите строку `strong`, иначе — `weak`.

Вторая задача

Ввод [23]: `password = 'Qwerty123'`

```
Ввод [24]: k = password.lower().count("andrei")
if k == 0:
    uq = set(password)
    lower_flag = False
    upper_flag = False
    digit_flag = False
    for char in uq:
        if char.islower():
            lower_flag = True
        if char.isupper():
            upper_flag = True
        if char.isdigit():
            digit_flag = True
    if len(uq) >= 4 and lower_flag and upper_flag and digit_flag:
        print('strong')
    else:
        print('weak')
else:
    print('weak')
```

strong

Решение второй задачи

Числа Фибоначчи

Как известно, [числа Фибоначчи](#) — это последовательность чисел, каждое из которых равно сумме двух предыдущих (первые два числа равны 1):
1, 1, 2, 3, 5, 8, 13, ...

Задание:

- 1) Определите число `amount` — количество чисел Фибоначчи, которые надо вывести;
- 2) Напишите код, который выводит первые `amount` чисел Фибоначчи.

Пример 1:

Input: 3

Output: 1 1 2

Пример 2:

Input: 10

Output: 1 1 2 3 5 8 13 21 34 55

Третья задача

Ввод [25]: `amount = 10`

Ввод [26]:

```
a = 1
b = 1
for i in range(amount):
    print(a, end=" ")
    c = a
    a = b
    b = b + c
```

1 1 2 3 5 8 13 21 34 55

Решение третьей задачи

На сайте <https://www.kaggle.com/> выберите любой набор данных в формате CSV и проведите для него маленькое исследование: загрузите данные из набора с использованием стандартного модуля `csv`, посмотрите средние значения и стандартные отклонения двух выбранных числовых атрибутов, найдите [методом наименьших квадратов](#) уравнение линейной зависимости, связывающей один числовой атрибут с другим. Для оценки заданной зависимости найдите [коэффициент парной корреляции](#), сделайте соответствующие выводы.

Результаты надо обязательно прокомментировать и пояснить!

Пример 1:

Пусть таблица `bikes.csv` содержит данные по арендам велосипедов за 2 года:

- `datetime` : дата и время аренды
- `season` : время года
- `temp` : температура воздуха по Цельсию
- `windspeed` : скорость ветра
- `registered` : число аренд

Одно из направлений исследования могло бы заключаться в проверке зависимости суммарного числа аренд от температуры воздуха.

Четвертое задание

В результате анализа сайта `kaggle.com` были выбраны данные по криптовалюте биткоин.

Объём - независимая - X

Цена закрытия - зависимая - Y

Зависимости цены закрытия от объёма.

Зависимость, которую требуется найти

Ввод [17]: # Функция для определения является ли строка вещественным числом

```
def is_number(str):  
    try:  
        float(str)  
        return True  
    except ValueError:  
        return False
```

Ввод [48]: # Считываем данные из файла
Считываются только числовые значения
4 столбец - цена закрытия
6 столбец - объём торгов

```
data = open('dataset.csv', 'r')  
df = []  
for line in data:  
    temp = line.replace('\n', '')  
    temp = temp.split(',')  
    if is_number(temp[4]):  
        df.append((float(temp[4]), float(temp[6])))
```

Считывание данных

Уравнение параболической регрессии имеет вид:

$$y = a_0 + a_1 \cdot x + a_2 \cdot x^2$$

Для нахождения коэффициентов параболы необходимо решить систему уравнений:

$$\begin{cases} na_0 + a_1 \sum_{i=1}^n x_i + a_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n y_i \\ a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 + a_2 \sum_{i=1}^n x_i^3 = \sum_{i=1}^n y_i x_i \\ a_0 \sum_{i=1}^n x_i^2 + a_1 \sum_{i=1}^n x_i^3 + a_2 \sum_{i=1}^n x_i^4 = \sum_{i=1}^n y_i x_i^2 \end{cases}$$

Для решение данной системы уравнений нам необходимо вычислить следующие значения:

$$\begin{aligned} \sum x_i \\ \sum y_i \\ \sum x_i^2 \\ \sum x_i^3 \\ \sum x_i^4 \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{aligned}$$

Данные необходимые для анализа

Ввод [49]: # Вывод системы уравнений, которую необходимо решить

```
print(f"a0*{n}+a1*{sum_x}+a2*{sum_x2}={sum_y}")  
print(f"a0*{sum_x}+a1*{sum_x2}+a2*{sum_x3}={sum_yx}")  
print(f"a0*{sum_x2}+a1*{sum_x3}+a2*{sum_x4}={sum_yx2}")
```

```
a0*2690+a1*39408335808274.0+a2*1.6619030512691775e+24=29813155.465381928  
a0*39408335808274.0+a1*1.6619030512691775e+24+a2*1.2322908569016729e+35=1.0677419732345071e+18  
a0*1.6619030512691775e+24+a1*1.2322908569016729e+35+a2*2.0227571563617705e+46=5.647841473167303e+28
```

Система уравнений, которую необходимо решить

```
Ввод [31]: # Функция для нахождения определителя матрицы
def determinant(matrix):
    if len(matrix) == 1:
        return matrix[0][0]
    else:
        n = len(matrix)
        summa = 0
        for i in range(n):
            minor = []
            for x in range(n):
                temporary = []
                for y in range(n):
                    if x != i and y != 0:
                        temporary.append(matrix[x][y])
                if len(temporary) == n - 1:
                    minor.append(temporary)
            if i % 2 == 0:
                summa += matrix[i][0] * determinant(minor)
            else:
                summa -= matrix[i][0] * determinant(minor)
        return summa
```

Функция расчета определителя

```

Ввод [32]: # Функция для решения СЛАУ
def slae_solver(x, y):
    det1 = determinant(x)
    if det1 == 0:
        return None
    n = len(x)
    ans = []
    deters = []
    for k in range(n):
        matrix = []
        for i, line in enumerate(x):
            temp = []
            for j, var in enumerate(line):
                if j == k:
                    temp.append(y[i])
                else:
                    temp.append(var)
            matrix.append(temp)
        deters.append(determinant(matrix))
    for det in deters:
        ans.append(det/det1)
    return ans

```

Функция для решения системы линейных уравнений

```

Ввод [13]: # Коэффициенты для решения уравнений
temp_x = [
    [n, sum_x, sum_x2],
    [sum_x, sum_x2, sum_x3],
    [sum_x2, sum_x3, sum_x4]
]
temp_y = [
    sum_y,
    sum_yx,
    sum_yx2
]

```

```

Ввод [14]: # Вывод уравнения параболической регрессии
ans = slae_solver(temp_x, temp_y)
print(f"y = {ans[0]} + {ans[1]}x + {ans[2]}x^2")

```

$y = 1187.896296047084 + 7.560368116947179e-07x + -1.911325770321666e-18x^2$

Получившиеся уравнений параболической регрессии

Out[15]: 0.7327908953378983

Подсчет коэффициента парной корреляции

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2; \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n. \end{cases}$$

Функция определения определителя, для решения СЛАУ


```
# Функция для решения СЛАУ

def slae_solver(x, y):
    det1 = determinant(x)
    if det1 == 0:
        return None
    n = len(x)
    ans = []
    deters = []
    for k in range(n):
        matrix = []
        for i, line in enumerate(x):
            temp = []
            for j, var in enumerate(line):
                if j == k:
                    temp.append(y[i])
                else:
                    temp.append(var)
            matrix.append(temp)
        deters.append(determinant(matrix))
    for det in deters:
        ans.append(det/det1)
    return ans
```

Функция для решения СЛАУ

Ввод [3]: *# Считываем данные*

```
n = int(input())
a = [list(map(float, input().split())) for i in range(n)]
b = list(map(float, input().split()))

3
2.0 -1.0 0.0
-1.0 1.0 4.0
1.0 2.0 3.0
0.0 13.0 14.0
```

Ввод [4]:

```
answer = slae_solver(a, b)
for elem in answer:
    print(elem, end = " ")
```

```
1.0 2.0 3.0
```

Пример работы программы

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Как осуществляется запуск Jupyter notebook?

Jupyter Notebook входит в состав Anaconda. Для запуска Jupyter Notebook перейдите в папку Scripts (она находится внутри каталога, в котором установлена Anaconda) и в командной строке наберите:

`ipython notebook`

2. Какие существуют типы ячеек в Jupyter notebook?

Markdown – для текста

Code – для python кода

Заголовок – для заголовков

Необработанный NBConwert – для того, чтобы блокнот не обрабатывал содержание ячейки

3. Как осуществляется работа с ячейками в Jupyter notebook?

Для запуска ячейки используйте команды из меню Cell, либо следующие сочетания клавиш:

Ctrl+Enter – выполнить содержимое ячейки.

Shift+Enter – выполнить содержимое ячейки и перейти на ячейку ниже.

Alt+Enter – выполнить содержимое ячейки и вставить новую ячейку ниже.

4. Что такое "магические" команды Jupyter notebook? Какие "магические" команды Вы знаете?

Для работы с переменными окружения используется команда `%env`.

Запуск Python кода из “.py” файлов, а также из других ноутбуков – файлов с расширением “.ipynb”, осуществляется с помощью команды `%run`.

`%%time` позволяет получить информацию о времени работы кода в рамках одной ячейки.

`%timeit` запускает переданный ей код 100000 раз (по умолчанию) и выводит информацию среднем значении трех наиболее быстрых прогонов.

5. Самостоятельно изучите работу с Jupyter notebook и IDE PyCharm и Visual Studio Code.

Приведите основные этапы работы с Jupyter notebook в IDE PyCharm и Visual Studio Code.

Jupyter notebook может работать только в профессиональной версии PyCharm.

1. Сначала вы должны создать новый проект.

2. В этом проекте создайте новый файл `ipynb`, выбрав File> New...> Jupyter Notebook. Это должно открыть новый файл записной книжки.

3. Если у вас не установлен пакет Jupyter Notebook, над вновь открытым файлом `ipynb` появится сообщение об ошибке. Сообщение об ошибке гласит: «Пакет Jupyter не установлен», и у вас будет опция «Установить пакет jupyter» рядом с ним.

4. Нажмите «Установить пакет jupyter». Это запустит процесс установки, который вы можете просмотреть, щелкнув запущенные процессы в правом нижнем углу окна PyCharm.

5. Чтобы начать изучение Jupyter Notebook в PyCharm, создайте ячейки кода и выполните их.

6. Выполните ячейку кода, чтобы запустить сервер Jupyter. По умолчанию сервер Jupyter использует порт 8888 по умолчанию на локальном хосте. Эти конфигурации доступны в окне инструментов сервера. После запуска вы можете просмотреть сервер над окном исходного кода, а рядом с ним вы можете просмотреть ядро, созданное как «Python 2» или «Python 3».

7. Теперь вы можете получить доступ к вкладке переменных в PyCharm, чтобы увидеть, как значения ваших переменных меняются при выполнении ячеек кода. Это помогает при отладке. Вы также можете установить точки останова в строках кода, а затем щелкнуть значок «Выполнить» и выбрать «Debug Cell» (или использовать сочетание клавиш Alt+Shift+Enter), чтобы начать отладку.

PyCharm предлагает три режима просмотра для редактирования файлов записной книжки Jupyter:

1. Режим «Только редактор»

Это позволяет добавлять и редактировать ячейки записной книжки.

2. Режим разделенного просмотра Режим разделенного просмотра позволяет добавлять ячейки и предварительно просматривать их вывод. Это также режим просмотра по умолчанию для всех записных книжек Jupyter в PyCharm.

3. Режим только предварительного просмотра

Здесь вы можете предварительно просмотреть результаты выполнения кода, необработанные ячейки и уценку кода.

Для начала, расскажем, как начать работу с Jupyter в VS Code.

- Если у вас еще нет существующего файла Jupyter Notebook, откройте VS Code Command Palette с помощью сочетания клавиш CTRL+SHIFT+P (Windows) или Command+SHIFT+P (macOS) и запустите команду «Python: Create Blank New Jupyter Notebook».

- Если у вас уже есть файл Jupyter Notebook, это так же просто, как просто открыть этот файл в VS Code. Он автоматически откроется с новым нативным редактором Jupyter.

Открыв Jupyter Notebook, вы можете добавлять новые ячейки, писать код в ячейках, запускать ячейки и выполнять другие действия с блокнотом.

VS code позволяет:

1. Обозревать переменные
2. Подключаться к удаленным серверам Jupyter
3. Экспортировать код в виде кода Python
4. Отладка кода