



*** Строки. Методы и функции.**

Использование срезов.

*** Кортеж. Основные операции с кортежем.**

Распаковка кортежа.

*** Список. Основные операции со списком.**

*** Словарь. Основные операции со словарем.**

*** Множества. Основные операции с множеством**

Турашова Анна Николаевна

Преподаватель

anna1turashova@gmail.com

Telegram: @anna1tur



Поверка домашнего задания



Задача 1

Создайте пустой файл `pyramid.txt` и напишите функцию, которая будет рисовать в этом файле пирамиду.

Функция принимает один аргумент – количество строчек в пирамиде.

```
#  
###  
#####  
#####
```

```
#  
###  
#####  
#####  
#####  
#####  
#####
```

```
#  
###  
#####  
#####  
#####  
#####  
#####  
#####  
#####  
#####
```



Задача 2

Дан файл с датами:

dates.txt		
1	2012/09/18	12:10
2	2079/01/21	09:09
3	2001/01/01	17:80
4	1968/17/19	02:21
5	1988/02/29	03:32
6	1012/06/18	19:10
7	2012/07/21	25:21
8	2021/05/31	01:12

2012/09/18 12:10
2079/01/21 09:09
2001/01/01 17:80
1968/17/19 02:21
1988/02/29 03:32
1012/06/18 19:10
2012/07/21 25:21
2021/05/31 01:12

Выбрать **существующие** даты в промежутке с 1950 по 2050 год.
Ответ вывести в любом формате.

Примеры:

2012/09/18 12:10 — Да

2079/01/21 09:09 — Нет (после 2050)

2001/01/01 17:30 — Да

1968/17/19 02:21 — Нет (17 месяца нет)



Работа с файлами



Чтение из файла

- Для чтения используется метод `read`

```
f = open("Толстой.txt", encoding="utf8")
text = f.read() # весь текст
print(len(text))
first = f.read(100) # первые 100 символов
print(first)
```

- Для чтения одной строки есть метод `readline()`

```
f = open("Толстой.txt", encoding="utf8")
for i in range(10):
    print(f.readline())
```

- Для чтения файла в список строк `readlines()`

```
lines = f.readlines()
```

Запись в файл



- Для записи в файл есть два режима: `w` (если файл существовал, его содержимое будет потеряно) и `a` — запись идет в конец файла. После выбора режима можно также ввести и символ `+`, который позволяет делать и операции чтения, и операции записи
- Метод `write` используется для записи, при этом перевод строки не добавляется автоматом, возвращает количество записанных символов

```
f = open("example.txt", 'w')
```

```
print(f.write('123\n456'))
```

```
f.close()
```

```
f = open("example.txt", 'r')
```

```
print(f.read())
```

- В файл может писать функция `print`, если ей передать именованный параметр `file`

```
print("Hello", file = f)
```



Заккрытие файла, блок with

- После работы с файлом он должен быть закрыт `f.close()`
- После завершения программы файлы автоматически закрываются
- Для того чтобы файл закрывался автоматически даже в случае ошибок в языке Python есть блок with

```
with open('Толстой.txt', 'r') as f:  
    read_data = f.read()  
print(read_data[:100])
```




Задачи



Задача 1.

Николай написал функцию `is_alive(health)`, которая проверяет здоровье персонажа в игре. Если оно равно или меньше нуля, то функция возвращает **False**, в противном случае **True**. К сожалению, функция не работает, так как ученик допустил в ней ряд ошибок. Исправьте их и проверьте работоспособность программы (в качестве аргумента всегда передается число).

```
def is_alive(health):  
    if:  
        health < 0  
        False  
    else:  
        return true
```



Задача 2.

Написать функцию `month_to_season()`, которая принимает 1 аргумент - номер месяца - и возвращает название сезона, к которому относится этот месяц.

Например, передаем 2, на выходе получаем 'Зима'.



Задача 3.

Иван решил создать самый большой словарь в мире.

Для этого он придумал функцию `biggest_dict(**kwargs)`, которая принимает неограниченное количество параметров «ключ: значение» и обновляет созданный им словарь `my_dict`, состоящий всего из одного элемента «`first_one`» со значением «`we can do it`».

Воссоздайте эту функцию.



Задача 4.

Дана строка в виде случайной последовательности чисел от 0 до 9.

Требуется создать словарь, который в качестве ключей будет принимать данные числа (т. е. ключи будут типом `int`), а в качестве значений – количество этих чисел в имеющейся последовательности.

Для построения словаря создайте функцию `count_it(sequence)`, принимающую строку из цифр.

Функция должна вернуть словарь из 3-х самых часто встречаемых чисел.



Задача 5.

Напишите функцию `tpl_sort()`, которая сортирует кортеж, состоящий из целых чисел по возрастанию и возвращает его.

Если хотя бы один элемент не является целым числом, то функция возвращает исходный кортеж.



Задача 6.

Функция `slicer()` на вход принимает кортеж и случайный элемент.

Требуется вернуть новый кортеж, начинающийся с первого появления элемента в нем и заканчивающийся вторым его появлением включительно.

Если элемента нет вовсе – вернуть пустой кортеж.

Если элемент встречается только один раз, то вернуть кортеж, который начинается с него и идет до конца исходного.



Задача 7.

Перед студентом стоит задача: на вход функции `sieve()` поступает список целых чисел.

В результате выполнения этой функции будет получен кортеж уникальных элементов списка в обратном порядке.



Задача 8.

Николай знает, что кортежи являются неизменяемыми, но он с этим не готов соглашаться.

Ученик решил создать функцию `del_from_tuple()`, которая будет удалять первое появление определенного элемента из кортежа по значению и возвращать кортеж без одного.

Попробуйте повторить шедевр не признающего авторитеты начинающего программиста.

К слову, Николай не всегда уверен в наличии элемента в кортеже (в этом случае кортеж вернется функцией в исходном виде).



Задача 9.

На основании 3 исходных множеств (передаются в качестве аргументов функции `diff()`) требуется написать функцию, которая будет возвращать либо симметричную разность, либо просто разность (если дополнительный аргумент функции `symmetric` имеет значение `False`) приведенных объектов в порядке: 1-ое множество, 2-ое множество, 3-е множество.



Задача 10.

Предоставлен список натуральных чисел.
Требуется сформировать из них множество.

Если какое-либо число повторяется, то преобразовать его в строку по образцу:

например, если число 4 повторяется 3 раза, то в множестве будет следующая запись:

- само число 4,

- строка «44» (второе повторение, т.е. число дублируется в строке),

- строка «444» (третье повторение, т.е. строка множится на 3).

Реализуйте вывод множества через функцию `set_gen()`.



Задача 11.

Требуется определить индексы первого и последнего вхождения буквы в строке.

Для этого нужно написать функцию `first_last(letter, st)`, включающую 2 параметра:

`letter` – искомый символ,
`st` – целевая строка.

В случае отсутствия буквы в строке, нужно вернуть кортеж `(None, None)`, если же она есть, то кортеж будет состоять из первого и последнего индекса этого символа.



Задача 12.

Николай решил вспомнить старые времена.

В свое время было модно писать сообщения с чередующимися заглавной и малой буквами.

Он захотел изобрести функцию, которая будет делать с любой предоставленной строкой аналогичное.

Ваша задача: повторить труд студента `camel(st)` с учетом того, что пробелы и знаки препинания не должны портить чередование регистра символов (они в этом процессе не учитываются, но возвращаются в итоговой строке).



Домашнее задание

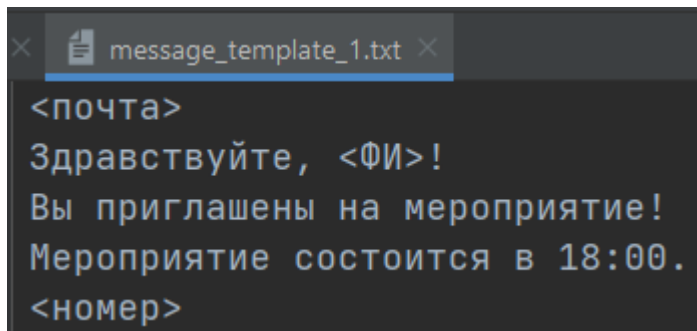


Вам дан файл-шаблон с неизвестным заранее содержимым и неизвестным количеством строк. А также файл с почтами, именами и фамилиями.

Вам необходимо написать алгоритм, который будет создавать новые файлы на основе шаблона. В новом файле на первой строке должна быть указана почта адресата, а на последней – номер файла в формате «номер_текущего/кол-во_всех». Где-то по центру шаблона необходимо вставить Фамилию и Имя адресата.

Для сгенерированных файлов создайте отдельную папку в корне проета.

Пример файла-шаблона:



```
<почта>  
Здравствуйте, <ФИ>!  
Вы приглашены на мероприятие!  
Мероприятие состоится в 18:00.  
<номер>
```

Пример файла с именами:



personal_data.txt

```
BreakInvention@gmail.com Калинин Кирилл  
MemoryUniverse@yandex.ru Зимина Амина  
SealSidewalk@mail.ru Уткина Анна  
MineBrokenFire@gmail.com Андреев Никита
```

Примеры сгенерированного письма:

```
BreakInvention@gmail.com  
Здравствуйте, Калинин Кирилл!  
Вы приглашены на мероприятие!  
Мероприятие состоится в 18:00.  
1/10
```

SealSidewalk@mail.ru



Уважаемый(ая) Уткина Анна!

Благодарим Вас за плодотворное сотрудничество и вклад в развитие нашей компании.
Приглашаем Вас на новогодний вечер, который мы устраиваем в честь наших партнеров, сотрудников и друзей.

В преддверии Нового 2021 года мы соберемся вместе, чтобы приятно провести время в праздничной атмосфере и пожелать друг другу успехов и новых побед.

Пусть наступающий год принесет Вам только приятные перемены, каждый день будет успешным в личной жизни и плодотворным в работе!

Ждём вас 30 декабря в 104 аудитории в 17:00.

3/10



Входит в ГК Аплана



АКАДЕМИЯ АЙТИ

Основана в 1995 г.

Е-learning
и очное
обучение

Направления обучения:

Информационные технологии

Информационная безопасность

ИТ-менеджмент и управление проектами

Разработка и тестирование ПО

Гос. и муниципальное управление

Филиалы:

Санкт-Петербург, Казань, Уфа, Челябинск,
Хабаровск, Красноярск, Тюмень, Нижний
Новгород, Краснодар, Волгоград, Ростов-на-Дону



Ежегодные награды
Microsoft,
Huawei, Cisco и
другие

Головной офис
в Москве

Разработка
программного
обеспечения и
информационных
систем

Программы по
импортозамещению

Ресурсы более 400
высококласных
экспертов и
преподавателей

Сеть региональных учебных центров
по всей России

Крупные заказчики



100+

сотрудников



АКАДЕМИЯ АЙТИ



Спасибо за внимание!

Центральный офис:

Москва, Варшавское шоссе 47, корп. 4, 7 этаж

Тел: +7 (495) 150-96-00

academy@it.ru

academyit.ru