

Задание 1. Введение в язык Python

Практикум 317 группы 2017-2018, осенний семестр

Начало выполнения задания: 6 сентября 2017 года.

Срок сдачи: 12 сентября 2017 года, 23:59.

Решение каждой задачи должно быть описано в модуле `task_<номер задачи>.py`.

В систему anytask необходимо сдать `zip` архив, содержащий решения задач, называющийся `contest1_<фамилия студента>_<имя студента>.zip`.

Задания проверяются с помощью автоматических тестов. Количество баллов за задачу зависит от количества пройденных тестов. Задание считается засчитанным, если хотя бы одна задача проходит больше 25% тестов.

1. Написать функцию `subreverse_sequence(sequence)`, принимающую список объектов Python. Функция должна возвращать кортеж, состоящий из элементов списка, стоящих на чётных местах, в обратном порядке (включая `A[0]`), после которых идут элементы списка, стоящие на нечётных местах. Использовать циклы запрещено!

Пример:

```
>>> subreverse_sequence([0, 1, 2, 3, 4, 5, 6])
(6, 4, 2, 0, 1, 3, 5)
```

2. Написать функцию `find_max_substring_occurrence(input_string)`, принимающую на вход непустую строку `input_string`. Функция должна возвращать наибольшее число `k`, такое что `input_string` совпадает с некоторой своей подстрокой `t`, выписанной `k` раз подряд.

Пример:

```
>>> find_max_substring_occurrence('abab')
2
```

3. По кругу записано несколько букв (возможно, повторяющихся). Студент 317 группы хочет узнать, сможет ли он прочитать некоторое слово, если будет двигаться по кругу (в любом направлении), не пропуская буквы. Студент сам выбирает место, с которого он начинает читать, и направление. Необходимо написать функцию `find_word_in_circle(circle, word)`, которая должна возвращать строку `'YES'`, если студент может найти строку `word` в круговой строке `circle`, и строку `'NO'` иначе. Строка `word` содержит как минимум один символ.

Пример:

```
>>> find_word_in_circle('napo', 'ap')
YES
```

4. В файле `input_dict_name` находится человеко-драконий словарь. В первой строке словаря записано число слов, к которым есть перевод. Затем на каждой строке словаря располагается слов и один или несколько переводов к нему. Слово и его переводы разделены дефисом, переводы одного слова разделены запятой. Дефис отделён от соседних слов пробельными символами. После запятой ставится пробельный символ. Функция `get_new_dictionary(input_dict_name, output_dict_name)` должна по человеко-драконьему словарю, находящемуся в `input_dict_name`, построить драконе-человечий словарь и сохранить его в файл с именем `output_dict_name` в аналогичном исходному словарю формате. Словарь должен быть полным, т.е. учитывать всю информацию, которая находилась в исходном словаре.

Пример:

Файл `'input.txt'`:

```
5
cat - kosha
dog - soba
good - horo, normo
bad - ploh, uzha
ugly - uzha
```

```
>>> get_new_dictionary('input.txt', 'output.txt')
```

Файл 'output.txt'

6

normo - good

kosha - cat

horo - good

uzha - bad, ugly

soba - dog

ploh - bad