

Во всех задачах необходимо тщательно проработать программу как с точки зрения модуля (выделить весь содержательный код в удобные функции), так и с точки зрения скрипта (сделать удобный интерфейс, написать описание, убедиться, что программа устойчива к некорректному вводу).

Также нужно самостоятельно разобраться с функцией `add_argument` класса `argparse.ArgumentParser` — многие ее параметры могут быть полезными при решении задач ниже.

1. На основе класса `Table` с прошлого занятия напишите утилиту-модуль для работы с таблицами.

Будем считать, что таблица записывается в файле следующим образом: строки отделены переносом `\n`, а столбцы — специальным разделительным символом или последовательностью символов (например, `\t` или `;`). Утилита должна получать через консоль тип операции с таблицей, файл(ы) с таблицей(ами) и все необходимые параметры, и выводить на экран результат обработки.

Утилита должна поддерживать операции выборки начала и конца таблицы (`head` и `tail`), операцию выборки столбцов (`cut`, список столбцов нужно получать из аргументов командной строки), операцию соединения таблиц по столбцам (`paste`). Вот несколько примеров вызовов утилиты:

```
$ python table.py cut -f 1,1,3,2 test.tsv
$ python table.py paste test1.tsv test2.tsv
$ python table.py head -n 2 test.tsv
```

Если число столбцов в какой-то строке меньше требуемого, то это считается ошибкой во входных данных, и программа должна сообщать об этом и завершать работу. Аналогично программа должна вести себя в любых других некорректных ситуациях.

Примечание 1. Можете выбрать любой способ нумерации столбцов (с нуля или с единицы).

Примечание 2. Для удобства работы с аргументами разберитесь с под-парсерами в библиотеке `argparse` (`parser.add_subparsers` и т.д.).

2. Напишите аналог Unix-утилиты `tr` (`translate`). Функция утилиты заключается в том, чтобы транслировать (заменять) символы из файла, используя таблицу замены. Таблица замены задается непосредственно в командной строке утилиты двумя строками; первая задает символы, которые заменяются, вторая — результаты замены (сопоставление идет по позиции). Например, после приведенного ниже вызова все символы `a` заменятся на символ `c`, а все символы `b` — на символ `d`.

```
$ python tr.py ab cd
```

Кроме того, утилита должна принимать опциональный список символов с ключом `-d` (`--delete`), которые нужно не заменять, а только удалить. Этот список должен быть более приоритетным, чем таблица замены, то есть, например, после приведенного ниже вызова строка `abcd` должна превратиться в строку `we`.

```
$ python tr.py abc qwe -d ad
```

Если в таблице есть ошибки (разные длины строк, повторяющиеся символы среди заменяемых), программа должна сообщать об этом и завершать работу. Можно считать, что замен глубины больше 1 (`a->b`, `b->c`) в аргументах нет.

Примечание. Использовать функцию `string.translate` не разрешается.