

Текстов трансформатор

В рамките на това домашно трябва да се разработи програма, която приема като вход текстов файл, извършва върху него различни трансформации и записва промените в нов файл във формат [Markdown](#).

Задачата ви е да реализирате **текстов трансформатор**. На командния ред ще ви се подаде път до текстов файл. В текстовия файл цялата информация е разделена на редове, като най-дългият ред, който може да се съдържа, е с по-малко от 1024 символа. Всеки ред от своя страна представлява последователност от думи, разделени от поне един символ за интервал или табулация.

Вашият текстов трансформатор трябва да зареди текста от файла, името на който ви е подадено от командния ред. След успешното зареждане на текста, потребителят трябва да може да изпълнява трансформации върху него, чрез следните команди:

- `makeHeading line` – прави подадения ред заглавен, където `line` е поредният номер на цял ред от вече заредения текст.
- `makeItalic line from to` – прави подадения интервал от думи (`from - to`), от даден ред (`line`), в *курсив (наклонен)*.
- `makeBold line from to` – прави подадения интервал от думи (`from - to`), от реда (`line`), **получер (удебелен)**.
- `makeCombine line from to` – от реда (`line`) прави подадения интервал от думи (`from - to`) **наклонен и удебелен**.
- `addLine lineContent` – добавя нов ред в заредения текст, чието съдържание е стойността на `lineContent`. При следваща команда за трансформация, този ред се взима предвид.
- `remove line` – премахва ред от вече заредения текст по неговия номер.
- `exit` – излиза от програмата

След приключване на програмата, промените се запазват в нов файл, със същото име, но с различно разширение (`.md`) и в различен формат (Markdown). Например ако оригиналният файл се е казвал `test.txt`, новият ще бъде с име `test.md`.

За повече информация как се реализират описаните по-горе трансформации в markdown формат вижте например:

- [md format](#)
- <https://guides.github.com/features/mastering-markdown/>
- <https://www.markdowntutorial.com/>

Към реализацията са наложени следните изисквания и ограничения:

- За съхранение на редовете и думите използвайте точното количество динамично заделена памет.
- Представете отделните елементи на системата, която реализирате, с класове.
- Следвайте добрите ООП практики за реализация на тези класове.
- При разработка на класовете можете (и е задължително) да добавите методи, които са необходими за правилното решение на задачата, но не са явно посочени в условието.
- Код, който не се компилира или грубо нарушава принципите на ООП, ще бъде оценен с нула точки.

Примерен вход:

Съдържание в blackHole.txt :

What Is a Black Hole?

This article is part of the NASA knows!

A black hole is a place in space where gravity pulls so much that even light can not get out. The gravity is so strong because matter has been squeezed into a tiny space. This can happen when a star is dying.

Команди:

makeHeading 1

makeBold 1 1 5

makeItalic 2 1 8

makeCombine 3 2 3

Изход: (съдържание в blackHole.md)

```
# **What Is a Black Hole?**
```

```
*This article is part of the NASA knows!*
```

A **black hole** is a place in space where gravity pulls so much that even light can not get out. The gravity is so strong because matter has been squeezed into a tiny space. This can happen when a star is dying.

И как се визуализира:

What Is a Black Hole?

This article is part of the NASA knows!

A **black hole** is a place in space where gravity pulls so much that even light can not get out. The gravity is so strong because matter has been squeezed into a tiny space. This can happen when a star is dying.