

## IML

Иван си е измислил език от тагове, но няма как да пише на него. Да се напише parser за езика IML (Ivan Markup Language). Езикът съдържа 8 tag-a: шест от тях имат и съответен затварящ и два нямат и са единични. Потребителят въвежда две имена на файлове. Първото име е за входен, а второто да изходен файл, като вашата програма трябва да parse-не съдържанието на входния файл в изходния. Ако във входния файл, участват и тагове извън езика да се изведе съобщение за грешка. Иван иска да се позволи влагане или вгнездяване на tag-ове. Т.е. във всеки произволен момент от четенето на файла, трябва да има повече отварящи, отколкото затварящи се tag-ове или броят им да бъде равен. Ако това не е така програмата извежда съобщение за грешка на стандартния изход за грешки.

1) `<trim>TEXT</trim>`

Ако някъде в TEXT има няколко whitespace-a един до друг -> оставя само първия.

2) `<toupper>TEXT</toupper>`

Извежда TEXT с главни букви.

3) `<tolower>TEXT</tolower>`

Извежда TEXT с малки букви.

4) `<hex>NUMBER</hex>`

Извежда в шестнайсетична бройна система записаното в десетична такава число NUMBER.

Забележка: буквите в шестнайсетичния запис зависят от tag-овете tolower и toupper, като по подразбиране са главни.

5) `<oct>NUMBER</oct>`

Извежда в осмична бройна система записаното в десетична такава число NUMBER.

6) `<pre > NUMBER</pre>`

Извежда NUMBER като число с два знака след десетичната точка.

7) `<nl>`

Извежда нов ред.

Забележка: това е единственият начин да се изведе нов ред. Т.е. новите редове в обикновения текст се игнорират

8) `<tb>`

Извежда 5 последователни space-a.

Ако някъде в tag-a toupper/tolower се среща tag tolower/toupper, то вътрешният tag override-ва външния. Същото важи и за tag-овете hex и oct.