PDF creator

The challenge

For this assessment, we'd like you to produce some C# or Java which takes input from a text file and produces a PDF output file. We'd recommend that you produce your solution in Visual Studio, Visual Studio Code, or IntelliJ. You can use third party libraries in your code to help wherever needed - for example, iText for C# and iText for Java are good options for helping create a PDF document.

The input file will contain a series of commands and text which describe what text should be in the output file and how it should be formated.

Example input:

.large

My PDF Document

.normal

.paragraph

This is my

.italic

very first

.regular

pdf document, and the output is formatted really well. While this paragraph is not filled, the following one has automatic filling set:

.paragraph

.indent +2

.fill

“Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.”

.paragraph

.nofill

.indent -2

Well that was

.bold

exciting

.regular

, good luck!

The above input text should be rendered roughly as follows:

My PDF Document

This is my *very first* pdf document, and the output is formatted really well. While this paragraph is not filled, the following one has automatic filling set:

> “Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.”

Well that was **exciting** , good luck!

In your solution you should repeat the above text over and over to make sure that it includes pagination, and should render at least 3 pages of content.

The commands are:

- **.paragraph:** starts a new paragraph
- **.fill:** sets indentation to fill for paragraphs, where the last character of a line must end at the end of the margin (except for the last line of a paragraph)
- **.nofill:** the default, sets the formatter to regular formatting
- **.regular:** resets the font to the normal font
- **.italic:** sets the font to italic
- **.bold:** sets the font to bold
- **.indent <number>:** indents the specified amount (each unit is probably about the length of the string "WWWW", but other values would work)

NOTES

At Gearset, we strive for good design and code that's well unit tested, or at least code that is written in a testable way. **We'll be closely assessing the quality of the code you've ritten in your solution.** This means we'll be expecting to see code which is easy for another developer to read through and digest;

which has sensible abstractions within it by splitting out classes and methods with distinct responsibilities; with well named classes, methods and variables.

Not got time to implement the refactorings you'd like to make your code as good as you'd like? Tell us about it! Are there some bits of your code that you're not particularly proud of but can't find a better approach? Tell us about it! As software engineers we have to make trade-offs between *time to get a job done* and *perfect code design all the time.* Identifying the trade-offs we're making is important.

Getting a good solution should take around half a day's work assuming you don't hit any significant roadblocks and depending on how many similar sorts of problem like this you've solved before.

What to submit

Send your solution and anything else useful for reviewing your solution - for example a supporting document detailing any assumptions you've made - over to us in a **.zip** archive. Your solution needs to be easy for us to load up in an IDE, build, and run - if it isn't then that'll make your solution more difficult to assess!

Ideally you'll send us a Visual Studio, Visual Studio Code, or IntelliJ project which we can load up and that reads from a text file which you've provided with your solution and then writes a new pdf to disk. We should then be able to open the PDF and see the output as we described above.