If something is wrong with this file or the transcriber, or you have a suggestion, please let me know at rogerhu (at) berkeley (dot) edu.

To begin the document: `..begin main`

To begin a section with heading **XYZ** with numbering: `..begin section number XYZ` To begin a section with heading **XYZ** without numbering: `..begin section XYZ`
So in this case, it's `..begin section number Creating the Document`

1 Creating the Document

Use `..begin preamble` and `..end preamble` to signal a part of the **preamble** to be ignored by the transcriber, including comments. The contents within them will automatically be inserted before the `\begin{document}` in the transcribed file. This is completely optional.

Use `..begin full preamble` and `..end full preamble` to completely override all default values in the preamble. The transcribed file will contain only the contents inside `..begin full preamble` and `..end full preamble` before `\begin{document}`.

Use `..begin ignore` and `..end ignore` to signal a part of the **document** to be ignored by the transcriber, including comments. The contents will be placed exactly where they were defined relative to everything else in the document.
`..begin ignore`
It's somewhat like `\verb` in LaTeX, where commands are ignored.
These are a couple of commands that will be ignored.
..begin main
..end main
However, doing something invalid in LaTeX like `\nl` would cause a compile-time error when using pdflatex because such a command is invalid. `..end ignore`

All of the following are optional.
`..name Your Name` will set the top right header to contain **Your Name**.
`..font XYZ` will set font to **XYZ**. Defaulted to 12
`..packages A B C` will include packages **A**, **B**, **C** in addition to amsmath, amssymb, amsthm, geometry, enumitem, and fancyhdr

`..paper letter` will set page size to letter. Can be “legal”, “a4”, and some others
`..orient landscape` will set page to be landscape. Default “portrait”

`..margin XYZ` will set margin size to **XYZ** inches. Default 1.
`..indent XYZ` will set paragraph indent to **XYZ**. Default 0, can be 4 (regular paragraph indent), 2, etc.
`..spacing XYZ` will set line spacing to **XYZ**. Default 1.5, can be “1”, “2”
`..assignment Template` will put **Template** at the left footer.
`..assetpath path/to/folder1 path/to/folder2` will set
`\graphicspath{{path/to/folder1}{path/to/folder2}}` in the preamble.

`..initeq XYZ ABC` will set the equation numbering scheme to be **subordinate** to counter type **ABC**. Equivalent to `\numberwithin{equation}{ABC}`
`..initheorem* XYZ` will create theorem type **XYZ** with no numbering. Equivalent to
`\newtheorem*{XYZ}{XYZ}`
`..initheorem XYZ` will create theorem type **XYZ** with a counter that increments each time **XYZ** is created. Equivalent to `\newtheorem{XYZ}{XYZ}`
`..initheorem XYZ ABC sub` will create theorem type **XYZ** that is **subordinate** to counter type **ABC**. Can be subordinate to equations. Equivalent to
`\newtheorem{XYZ}{XYZ}[ABC]`
`..initheorem XYZ ABC shared` will create theorem type **XYZ** that is **shared** with counter type **ABC**. Can be shared with equations. Equivalent to
`\newtheorem{XYZ}[ABC]{XYZ}`
`..qed XYZ` will change the symbol that sits at the end of a proof to be **XYZ**. By default, it is a white square.

2 Keywords

Line comment: `%`

Block comment: `%%% something here %%%`

Some of the following are honestly kind of useless like `\nl -> \\`.

`\union`	`->`	$\cup$
`\itsc`	`->`	$\cap$
`\setdiff`	`->`	$\setminus$
`\del`	`->`	∇
`\norm{\vec{a}}`	`->`	$\lVert\vec{a}\rVert$
`\floor{x}`	`->`	$\lfloor x\rfloor$
`\ceil{x}`	`->`	$\lceil x\rceil$
`\nin`	`->`	$\notin$
`\R`	`->`	$\mathbb{R}$
`\Z`	`->`	$\mathbb{Z}$
`\N`	`->`	$\mathbb{N}$
`\Q`	`->`	$\mathbb{Q}$
`\C`	`->`	$\mathbb{C}$
`\aand`	`->`	$\wedge$
`\oor`	`->`	$\vee$
`\<=`	`->`	$\leq$
`\>=`	`->`	$\leq$
`\<<`	`->`	$\ll$
`\>>`	`->`	$\gg$
`\~=`	`->`	$\approx$
`\<(`	`->`	$\langle$
`\>)`	`->`	$\rangle$
`\cross`	`->`	$\times$
`\nimplies`	`->`	$\nRightarrow$
`\ddef`	`->`	$\overset{\text{def}}{=}$
`\sset`	`->`	$\overset{\text{set}}{=}$
`\ital{a}`	`->`	*a*
`\bold{a}`	`->`	**a**
`\contradiction`	`->`	$\Rightarrow\Leftarrow$

```
\nl             ->          New line character
\tb             ->          Tab character
```

Check transcribe.py for what exactly they're changed to.

3 Copy Pasting

Note: This will not copy Theorem numbers correctly. Use `restatable` environment from `thm-restate` for that.

`..begin copy label1`
This text is between the `..begin copy label1` and `..end copy`. It will get duplicated when `..paste label1` is present.
`..end copy`

`..paste label1`
This text is between the `..begin copy label1` and `..end copy`. It will get duplicated when `..paste label1` is present.

`..begin copy label2`
Second set of copied text
`..end copy`

`..paste label2`
Second set of copied text

`..paste label2`
Second set of copied text

Do keep in mind that if a label is reused, `..paste label` will only use the last reference to that label, so the originally copied text under "label" will be lost, even if there are `..paste` commands before future declarations of the same label.

For example:
`..begin copy label3`
This is inside the first label3.
`..end copy`

`..paste label3`
This is inside the second label3.

`..begin copy label3`
This is inside the second label3.
`..end copy`

`..paste label3`

This is inside the second label3.

`..paste` commands can also be called before the copy reference is established.

`..paste label4`

This is label4

`..begin copy label4`

This is label4

`..end copy`

4 Theorems

To begin a type of theorem defined in the header, such as **XYZ**: `..begin thm XYZ`. When you're done with the theorem, `..end thm XYZ` must be used as well.

```
..begin thm Theorem
```

Theorem. *This is the contents of Theorem. Notice that it does not have a counter.*

```
.. end thm Theorem
```

That means it doesn't matter how many *Theorem*s you have, they'll always show up as **Theorem**.

Theorem. *Here is another theorem.*

Theorem. *And another.*

On the other hand, *Lemma*s have a counter, since it was defined without the asterisk.

```
..begin thm Lemma
```

Lemma 1. *Yes, I know this isn't a lemma, but see how there's a counter?*

```
.. end thm Lemma
```

And since *Note* shares the same counter as *Lemma*, the counter will change whenever we define a new *Note* or *Lemma*.

```
..begin thm Note
```

Note 2. *Hello World! New counter!*

```
..end thm Note
```

Lemma 3. *Counter changes again!*

However, *Corollary*s are defined to be subordinate to *Lemma* counters, take a look.

```
..begin thm Corollary
```

Corollary 3.1. *Hello, I'm a corollary.*

```
..end thm Corollary
```

Corollary 3.2. *What a coincidence, me too!*

Once the "superior" counter changes, in this case *Lemma*, *Corollary*s' counter will also change.

Note 4. *Our counter changed!*

Corollary 4.1. *It should be self-explanatory why my counter is the way it is now.*

To begin a theorem **XYZ** with specific heading **ABC DEF**: `..begin thm XYZ ABC DEF`. To end the theorem, using `..end thm XYZ` is enough.

```
..begin thm Theorem Stoke's Theorem
```

Theorem (Stoke's Theorem)**.** *For a closed surface oriented counter-clockwise,*

$$\int\limits_C \vec{F} \cdot d\vec{r} = \iint\limits_S (\nabla \times \vec{F}) \cdot d\boldsymbol{S}$$

```
..end thm Theorem
```

Note 5 (Some arbitrary title)**.** *It works for* Note, Lemma*, etc.*

Proofs

`..begin subsection Proofs`

To begin a subsection **XYZ**, use `begin subsection XYZ`. You can also have subsubsections, subsubsubsections, etc.

To create a proof, surround the contents of the proof with `..begin proof` and `..end proof`

Claim: an irrational number raised to an irrational power can be rational.

`..begin proof`

Proof. Notice that $\sqrt{2} \in \mathbb{R}, \notin \mathbb{Q}$. Therefore, we know that $\sqrt{2}^{\sqrt{2}} \in \mathbb{R}$ but can be $\in \mathbb{Q}$ or $\notin \mathbb{Q}$.

Suppose $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$, we're done.

Suppose $\sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$, we can then use this value and raise it to another power of $\sqrt{2}$.

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}*\sqrt{2}} = \sqrt{2}^{2} = 2 \in \mathbb{Q}$$

Q.E.D

`..end proof`

5 Lists

Lists can be nested.
Do this to create a bullet-point (unnumbered) type of list:
`..begin list bullet [optional_label]`.
To create a numbered list: `..begin list [optional_label]`
The optional label should not be surrounded by brackets. For bullet point lists, any character can be used as the label.
For numbered lists, use “1” without the quotes to number using arabic numerals, “a” without the quotes to use the English alphabet, and/or “i” without the quotes to use roman numerals.
Because of the way the transcriber is currently designed, it is impossible to have it use “1a”, “1b”, “1c” as consecutive labels. In the future, a backslash to escape the character or something like that will probably allow this.
Regardless of whether it is a numbered list, a list with special labels, use `..end list` at the very end.

This is the syntax for a sample list.

```
..begin list bullet --
.. item 1
.. item 2
This line will be appended to the end of item 2
..n So will this one, but we just added a new line character, so it visually looks like
it’s on a different line, but there just would not be a bullet point in front of it.
..begin list a)
.. Nested lists can be created.
..begin list (i)
.. So can nested lists inside nested lists.
.. abcdefg
..begin list .1.
.. Something here
.. Something else here
..end list
..end list
.. The above line must be used to indicate that a list is ended.
.. It'll transcribe correctly, but it will not compile to .pdf
..end list
.. an end list must be used for each list created.
..end list
```

Here is the above code compiled.

- item 1
- item 2 This line will be appended to the end of item 2 ..n So will this one, but we just added a new line character
    a) Nested lists can be created.
        (i) So can nested lists inside nested lists.
        (ii) abcdefg
            .1. Something here
            .2. Something else here
    b) The above line must be used to indicate that a list is ended.
    c) It'll transcribe correctly, but it will not compile to .pdf
- an end list must be used for each list.

Note 6. *The transcribed .tex files will automatically be indented wherever there are items in a list.*

6 Parts

The **Parts** and **ResumeParts** environment is heavily inspired by CS 70's header.sty file.

Usage of `..begin parts`

```
..begin parts [optional_label]
p\ Part 1 contents
p\ Part 2 contents
p\ Part 3 contents
..end parts
```

Usage of `..resume parts`

```
..resume parts
p\ Part 1 contents
p\ Part 2 contents
p\ Part 3 contents
..end parts
```

```
..begin parts
p\ Parts are kind of like lists, except they look much better
if you're not actually trying to make a short list.
p\ Precede each part with \verb|p\ |
p\ To end the parts environment, use \verb|..end parts|
..end parts
```

(a) Parts are kind of like lists, except they look much better if you're not actually trying to make a short list.

(b) Precede each part with "`p\ `".

(c) To end the parts environment, use `..end parts`

At any point in time, the parts environment can be resumed by using `..resume parts`

```
..resume parts
p\ These will continue the same labels from before.
p\ To end this environment, use \verb|..end parts| again.
..end parts
```

(d) These will continue the same labels from before.

(e) To end this environment, use `..end parts` again.

```
..begin parts i)
p\ Item 1
p\ Item 2 \nl
Some text here.
p\ ..begin proof
This is a proof environment.
..end proof
p\ Continuing on the parts environment.
..end parts
```

i) Item 1

ii) Item 2
Some text here.

iii) *Proof.* This is a proof environment. Q.E.D

iv) Continuing on the parts environment.

7 Math

Equations

LaTeX's equation mode honestly sucks. You can't have a blank line while in equation / gather / align mode. This means that a reasonable person expecting to create a new line would get a ridiculous error like missing `$` inserted. The only way to create a line break would be using \\, which I think is a lot less intuitive than it should be.

This transcriber will assume all new lines within the gather or align mode are supposed to be in a new line compared to the previous and following line. This means that there is no need to insert \\ at the end of each line, but it also means that the only way to continue on the same line in the compiled pdf is to put the equation in the same line. Blank lines will be automatically removed, so feel free to leave blank lines in the file to be transcribed.

Only two modes are implemented: **align** and **gather**. I honestly don't know what the difference between **gather** and **equation** are, so I'm only implementing **gather**. **Align** mode allows you to select a symbol, usually the equal sign, to be aligned to other equations; **gather** mode centers all the equations. Just like LaTeX's align, prepend the symbol or character you want to align with a & sign.

By default, **gather** mode, and non-numbering mode is used. Enable **align** by putting the word *align* in it. Enable **numbered** mode by putting *number* in it.

Here's the **gather** mode with no numbering:

```
..begin eq
a = b + c
d = e + f
a + d = b + c
..end eq
```

$$\begin{gather*} a = b + c \\ d = e + f \\ a + d = b + c \end{gather*}$$

Here's the **align** mode with no numbering:

```
..begin eq align
```

```
h &= 2x + 3y + 4z      &   2u + &2v = 10
x &= y                 &   &2v + 4w = 20
a &= b + c             &   2b + 2c + &7v = 30
f(x, y) &= g(a, b, c)  &   &2v = 4z
..end eq
```

$$\begin{aligned} h &= 2x + 3y + 4z & 2u+&2v = 10 \\ x &= y & &2v + 4w = 20 \\ a &= b + c & 2b + 2c+&7v = 30 \\ f(x, y) &= g(a, b, c) & &2v = 4z \end{aligned}$$

Equation numbering is subordinate to sections, but it can be changed with "..initeq"

Here's the **gather** mode with numbering:

```
..begin eq number
a = b
2x + 4c = 10
10 + 4c = 20
..end eq
```

$$a = b \tag{7.1}$$

$$2x + 4c = 10 \tag{7.2}$$

$$10 + 4c = 20 \tag{7.3}$$

Here's how to do **align** with numbering:

```
..begin eq align number
a &= b + c
a &= 2b + d
a &= 3c + e
..end eq
```

$$a = b + c \tag{7.4}$$

$$a = 2b + d \tag{7.5}$$

$$a = 3c + e \tag{7.6}$$

Matrices

To create a matrix, use `..begin matrix`. Normally, in amsmath, math mode must be enabled prior to declaring the matrix, this is not the case anymore.
Originally, a `&` must be used to separate values within the matrix. This transcriber will assume that spaces are the delimiter. There is also no need to use a line break character, because new lines are assumed to be part of the next row of values.
Here is the code for defining a basic matrix:

```
..begin matrix
a b c d
e f g h
i j k l
..end matrix
```

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix}$$

There are currently no customizations to what matrices are surrounded by – they'll always be brackets because I like the look of brackets more than parentheses. There are no plans to change this in the future.

Arrays and Tables

To create an array or table, use `..begin table math [formatting]` or `..begin table [formatting]`, respectively. The formatting for the table should be exactly what one would do in LaTeX's `\begin{tabular}{[formatting]}` or `\begin{array}{[formatting]}`. The only expected characters within the formatting are "c", "l", "r", and "|". The three letters signify what that column's alignment should be: center, left, or right; while the "|" character represents a line separation between them. **In LaTeX, a & is used as the delimiter between two columns. Here, a single \followed immediately by a space is to be used.** A line break is also automatically assumed to be on a separate line, although `\hline` would be an exception.

Here is the code for an array (math mode):

```
..begin table math r|c|l|
2 \ 3 \ 465
\hline
p \implies q \ q \implies r \ 6
..end table
```

2	3	4
$p \implies q$	$q \implies r$	6

Similarly, here is some sample code for a table (non math mode):

```
..begin table c|c|c
\ Names \
\hline
John \ Mary \ Jane
Adam \ Kevin \ Allie
..end table
```

	Names	
John	Mary	Jane
Adam	Kevin	Allie

8 Text Alignment

By default, all text are flushed left. However, it is possible to change it to be flushed right, centered, justified, etc. Use the following to set alignment.

`..align center`
This is a line of text that is centered.

`..align right`
This is a line of text that is flushed right.

`..align justify`
Lorem ipsum dolor sit amet, consectetur adipiscing 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.

`..align left`
Lorem ipsum dolor sit amet, consectetur adipiscing 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.

To end the document, use `..end main`
This is actually entirely optional, as long as there is at least 1 empty line at the end.