Extension

Due to work in the software engineering module, I came to realise the importance of testing and the great difficulty of adding testing into your code after it is finished. With this in mind and the I decided to take test driven approach to the design and implementation of my software in order to demonstrate an aspect of software engineering. This resulted in no segmentation faults when writing my parser, but ironically still had to contend with them in my testing functions. Adding functions, features and tests as I progressed was much easier than I had previously experienced. More about this is written in test.pdf.

As well as this I added a couple of extra features to the language, I have highlighted them in blue in the formal grammar my turtle program will now parse and interpret.

I added the ability to change the colour. I let the user input a word and interpreted that word into rgb. I only programmed 4 colours, it would be very easy to add more if needed.

I also added an if function, this enables the user to change colour, direction or even add whole new loops when a certain condition is met.

<MAIN>	:=	“{“ <INSTRCTLST>
<INSTRCTLST>	:=	<INSTRUCTION><INSTRCTLST> |
		"}"
<INSTRUCTION>	::=	<FD> |
		<LT> |
		<RT> |
		<DO> |
		<SET>|
		<IF>
<FD>	:=	"FD" <VARNUM>
<LT>	::=	"LT" <VARNUM>
<RT>	::=	"RT" <VARNUM>
<DO>	::=	"DO" <VAR> "FROM" <VARNUM> "TO"
		<VARNUM> "{" <INSTRCTLST>
<IF>	::=	“IF” <VAR> “:=” <VARNUM><MAIN> |
		“IF” “COLOUR” “:=” <COLOUR><MAIN>
<COLOUR>	::=	“RED” | “BLUE” | “GREEN” | “WHITE”
<VAR>	::=	[A-Z]
<VARNUM>	::=	number |
		<VAR>
<SET>	::=	"SET" <VAR> ":=" <POLISH>|
		“SET” “COLOUR” “:=” <COLOUR> “;”
<POLISH>	::=	<OP> <POLISH> |
		<VARNUM> <POLISH> |
		";"
<OP>	::=	"+" |
		"-" |
		"*" |
		"/"