Exercises for chapter: TEX – macro programming

1. Write a macro `\intt` ('in typewriter type') such that `\intt{foo}` and `\intt{foo_bar}` are output as `foo` and `foo_bar`, in typewriter type.
2. Write a macro that constructs another macro: `\tees\three3` should be equivalent to `\def\three{TTT}`, `\tees\five5` equivalent to `\def\five{TTTTT}` et cetera. In other words, the first argument of `\tees` is the name of the macro you are defining, the second is the number of letters 'T' the defined macro expands to. To make sure that your solution really expands to that string of 'T's, and not some code that generates it when the macro is called, do `\show\five` and check the screen output.
3. TEX natively has addition, multiplication, and division arithmetic. Write a square root routine in TEX. Hint: Use Newton's method.
4. Make this work:

```
\def\LeftDelim{(}\def\RightDelim{)}
\DefineWithDelims{foo}{My argument is `#1'.}
\def\LeftDelim{<}\def\RightDelim{>}
\DefineWithDelims{bar}{But my argument is `#1'.}
\foo(one)\par
\bar<two>
```

   Output:

   My argument is 'one'.

   But my argument is 'two'.

   In other words, `\DefineWithDelims` defines a macro – in this case `\foo` – and this macro has one argument, delimited by custom delimiters. The delimiters can be specified for each macro separately.
   Hint: `\DefineWithDelims` is actually a macro with only one argument. Consider this code snippet:

```
\Define{foo}{ ... #1 ...}
\def\Define#1{
  \expandafter\def\csname #1\endcsname##1}
```