Appendix D in the \TeX{}book has the task of defining $\ast$s as a macro containing $\text{number\textbackslash n}$ copies of an asterisk. The solutions in the TeXbook are not really fun. Here is one that is all of fun, efficient and simple:

\begin{verbatim}
def\asts#1{\if#1m*\expandafter\asts\fi}
def\asts{\expandafter\asts\romannumeral\number\number\number000\relax}
\end{verbatim}

Now for something more general: we want a macro $\replicate$ that gets a number in its first argument and arbitrary tokens in its second argument and expands to the given number of repeated token strings.

It is surprisingly hard to pass both the shrinking string of $m$ as well as the argument to repeated in a useful way into the expanding first macro, and the reader is advised to try it. What I came up with was

\begin{verbatim}
\long\def\gobble#1{}
\long\def\xii#1\#2{\if#2m#1\expandafter\xii\else\expandafter\gobble\fi\#1}
\long\def\xiii#1\relax\#2{\xii{#2}\#1\relax}
def\replicate#1{\expandafter\xiii\romannumeral\number\number\number#1\relax}
\end{verbatim}

A somewhat Wittier variant that takes its toll on the semantic nest size would be

\begin{verbatim}
\long\def\gobble#1{}
\long\def\xii#1\#2{\if#2m\expandafter\xii\else\expandafter\gobble\fi#1}
\long\def\xiii#1\relax#2{\xii{#2}\#1\relax}
def\replicate#1{\csname rn#1\replicate\endcsname}
\end{verbatim}

\begin{verbatim}
def\replicate#1\#2{\ifnum\number\#1>0 #2% \expandafter\replicate\expandafter{\number\numexpr\#1-1}{#2}\fi}
\end{verbatim}

Of course, if we are leaving the area of \TeX{} compatibility and take a look at what we can do with \texttt{e-\TeX{}}, we arrive at the boring