The `binhex.tex` package for expandable conversion into binary-based number systems

David Kastrup*

July 22, 2005

Contents

1 Usage

1 Usage

This is a file for expandably converting numbers into binary, octal and hexadecimal. All constructs \TeX's \texttt{number} primitive are valid. This holds for all numeric arguments of the macros presented in here.

You use this package by simply inputting it with

\begin{verbatim}
\input binhex
\end{verbatim}

It will work equally well under \LaTeX{} and plain \TeX. It does not even use plain \TeX, but only \TeX{} primitives. Simply setting the correct \texttt{catcode} values for \texttt{#} and end of line will make it load and work under \texttt{initex}.

The following macros are defined: \texttt{\binary{⟨number⟩}} will convert \texttt{⟨number⟩} into its binary representation.

\begin{verbatim}
\binary{0} → 0
\binary{\maxdimen} → 111111111111111111111111111111
\binary{-"7EE6} → -111111011100110
\end{verbatim}

\texttt{\binary{⟨size⟩}{⟨number⟩}} will convert \texttt{⟨number⟩} into a binary representation of at least \texttt{⟨size⟩} digits length, filling up with leading zeros where necessary. The - sign of negative numbers is not counted. If both \texttt{⟨size⟩} and \texttt{⟨number⟩} are zero, an empty string is generated. This should please some computer scientists in some situations.

\begin{verbatim}
\nbinary{3}{3} → 011
\nbinary{3}{-2} → -010
\nbinary{3}{-12} → -1100
\end{verbatim}

\texttt{\hex{⟨number⟩}} converts \texttt{⟨number⟩} into its hexadecimal representation, using

*David.Kastrup@neuroinformatik.ruhr-uni-bochum.de
uppercase letters.

```
\hex{34} \rightarrow 22
\hex{-4711} \rightarrow -1267
```

\textbf{\texttt{\textbackslash hex}} \hspace{1em} \texttt{\textbackslash hex{(size)}{(number)}} will convert \texttt{(number)} into a hexadecimal representation of at least \texttt{(size)} digits length, filling up with leading zeros where necessary. The \texttt{-} sign of negative numbers is not counted. If both \texttt{(size)} and \texttt{(number)} are zero, an empty string is generated. This should please some computer scientists in some situations.

```
\hex{3}{3} \rightarrow 003
\hex{3}{-\maxdimen} \rightarrow -3FFFFFFF
```

\textbf{\texttt{\textbackslash oct}} \hspace{1em} \texttt{\textbackslash oct{(number)}} converts \texttt{(number)} into its octal representation.

```
\oct{34} \rightarrow 42
\oct{-4711} \rightarrow -11147
```

\textbf{\texttt{\textbackslash noct}} \hspace{1em} \texttt{\textbackslash noct{(size)}{(number)}} will do the right thing.

```
\noct{3}{13} \rightarrow 015
\noct{3}{\maxdimen} \rightarrow -7777777777
```

\textbf{\texttt{\textbackslash tetra}} \hspace{1em} \texttt{\textbackslash tetra{(number)}} is for people counting with arms and legs instead of fingers, or for quadrupeds.

```
\tetra{34} \rightarrow 202
\tetra{-4711} \rightarrow -1021213
```

\textbf{\texttt{\textbackslash ntetra}} \hspace{1em} \texttt{\textbackslash ntetra{(size)}{(number)}} is for those of the same count which have minimum requirements.

```
\ntetra{3}{3} \rightarrow 003
\ntetra{3}{\maxdimen} \rightarrow -333333333333333
```

\textbf{\texttt{\textbackslash nbinbased}} \hspace{1em} \texttt{\textbackslash nbinbased{(logbase)}{(size)}{(number)}} will convert \texttt{(number)} into number base \texttt{2^{logbase}} and generate at least \texttt{(size)} digits. Only supported values of \texttt{(logbase)} are 1, 2, 3, 4. This is called by all other macros except of the faster binary conversion macros.

```
\nbinbased{3}{3}{13} \rightarrow 015
\nbinbased{3}{3}{\maxdimen} \rightarrow -7777777777
\nbinbased{2}{4}{13} \rightarrow 0031
\nbinbased{2}{4}{\maxdimen} \rightarrow -333333333333333
```

That’s it, have fun!