

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{ffcode}
\begin{document}
The function \texttt{fibo()} is recursive:
\begin{ffcode}
int fibo(int n) {
  if (n < 2) {
    return n; \label{ln:ret}
  }
  return fibo(n - 1) + fibo(n - 2);
}
\end{ffcode}
The line no. 3 terminates it.
\end{document}

You have to run \texttt{pdflatex} with the \texttt{-shell-escape} flag in order to let \texttt{minted} (the package we use) to run \texttt{Pygments} and format the code. If you don’t want this to happen, just use the \texttt{nopygments} option.

A pair of vertical lines decorate a \TeX{} command inside the snippet. If you want to print a single vertical line, use this: \texttt{\vert}.

If you want to omit the light gray frames around \texttt{\ff} texts, use the package option \texttt{noframes}.
The command \texttt{\textbackslash ff} behaves differently in math mode: it doesn’t add gray frames:

\[ x = \int_{\text{home}}^{N} f(x). \tag{1} \]

To omit the vertical gray bar at the left side of each snippet, use the \texttt{nobars} option of the package.

To omit the line numbers, use the \texttt{nonumbers} option of the package.

By default, the numbering is continuous: line numbers start at the first snippet and increment until the end of the document. If you want them to start from one at each snippet, use \texttt{nocn} (stands for “no continuous numbering”) option of the package.

You can highlight some lines in your \texttt{ffcode} environment, or can use any other additional configuration parameters from the \texttt{minted} package:

\begin{verbatim}
\begin{ffcode*}{highlightlines={1,4-5}}
while (true) {
    print("Hello!")
    print("Enter your name:")
    scan(x)
    print("You name is " + x)
}
\end{ffcode*}
\end{verbatim}

Using this second argument of the \texttt{ffcode*} (with the trailing asterisk), you can provide any other options from the \texttt{minted} package to the snippet.

By the way, the package correctly formats low-height texts, for example, just a dot: .

More details about this package you can find in the yegor256/ffcode GitHub repository.