`\aftergroup` ⟨*token*⟩

When TEX encounters this command during input, it saves ⟨*token*⟩. After the end of the current group, it inserts ⟨*token*⟩ back into the input and expands it. If a group contains several `\aftergroup`s, the corresponding tokens are *all* inserted following the end of the group, in the order in which they originally appeared.

The example that follows shows how you can use `\aftergroup` to postpone processing a token that you generate within a conditional test.

Example:

```
\def\neg{negative} \def\pos{positive}
% These definitions are needed because \aftergroup applies
% to a single token, not to a sequence of tokens or even
% to a brace-delimited text.
\def\arith#1{Is $#1>0$? \begingroup
   \ifnum #1>-1 Yes\aftergroup\pos
   \else No\aftergroup\neg\fi
   , it's \endgroup. }
\arith 2
\arith {-1}
```

produces:

Is $2 > 0$? Yes, it's positive. Is $-1 > 0$? No, it's negative.