**Tokenising**

MicroBASIC is run in tokenised form. The following things are legal tokens.

1. Known keywords are tokenised using character code 128 onwards. These can be any ASCII characters. When decoded, the presence of a leading or trailing alphanumeric defines whether a space is inserted before or after (removing duplicates). E.g 4 AND 5 would be tokenised as ‘4’[AND]’5’ and decoded as 4 and 5 irrespective of spaces. RND( 32 would remove the space, it would tokenise as [RND(]32.
2. Keywords are detokenized in lower case.
3. Other than that, tokens are encoded in standard ASCII, converted to lower case
4. Spaces are ignored.
5. A tokenised strings end with C’s string terminator, chr(0)
6. Strings begin and end with “”, character code 34.
7. Once a REM token has been found, the tokenising copies the rest of the string out.

*BYTE8 TOK\_TokeniseString(BYTE8 \*input)*

Tokenises a string and returns a pointer to the string

*BYTE8 TOK\_DeTokeniseString(BYTE8 \*input)*

Detokenises a string and returns a pointer to the string

Both of these use a fixed length buffer to store the result, failing if there is an error (e.g. overflow).

To speed up interpreting, keywords should be placed in groups.

1. Term operators
2. Expression operators
3. Language keywords