4.2.1 Character string

4.2.1.1 Single keywords

A character string value *shall* be composed only of the set of restricted ASCII text characters, decimal 32 through 126 (hexadecimal 20 through 7E) enclosed by single quote characters (“’”, decimal 39, hexadecimal 27). A single quote is represented within a string as two successive single quotes, e.g., O’HARA = `'O''HARA'`. Leading spaces are significant; trailing spaces are not. This standard imposes no requirements on the case sensitivity of character string values unless explicitly stated in the definition of specific keywords.

If the value is a fixed-format character string, the starting single quote character *must* be in byte 11 of the keyword record and the closing single quote *must* occur in or before byte 80. Earlier versions of this standard also *required* that fixed-format characters strings *must* be padded with space characters to at least a length of eight characters so that the closing quote character does not occur before byte 20. This minimum character string length is no longer required, except for the value of the `XTENSION` keyword (e.g., `'IMAGE␣␣␣'` and `'TABLE␣␣␣'`; see Sect. 7) which *must* be padded to a length of eight characters for backward compatibility with previous usage.

Free-format character strings follow the same rules as fixed-format character strings except that the starting single quote character *may* occur after byte 11. Any bytes preceding the starting quote character and after byte 10 *must* contain the space character.

Note that there is a subtle distinction between the following three keywords:

```
KEYWORD1= ''               / null string keyword
KEYWORD2= '   '            / empty string keyword
KEYWORD3=                  / undefined keyword
```

The value of `KEYWORD1` is a null, or zero length string whereas the value of the `KEYWORD2` is an empty string (nominally a single space character because the first space in the string is significant, but trailing spaces are not). The value of `KEYWORD3` is undefined and has an indeterminate data type as well, except in cases where the data type of the specified keyword is explicitly defined in this standard.

The maximum length of a string value that can be represented on a single keyword record is 68 characters, with the opening and closing quote characters in bytes 11 and 80, respectively. In general, no length limit less than 68 is implied for character-valued keywords.

4.2.1.2 Continued string keywords

Earlier versions of this Standard only defined single string keywords as described in the previous section. The Standard now incorporates a convention (originally developed for use in *FITS* files from high energy astrophysics missions) for continuing arbitrarily long string values over multiple consecutive keyword records. This convention is mainly intended for use with new user-defined keywords; it *must not* be used with the `XTENSION`, `TFORMn`, `EXTNAME`, `TTYPEn`, `TDISPn`, and `TNULLn` keywords, and it is *not recommended* for use with any other string-valued keywords that are defined in this Standard.

Long keyword string values can be represented in *FITS* headers by continuing the string over multiple keyword records using the following procedure:

1. Divide the long string value into a sequence of smaller substrings, each of which is no longer than 67 characters in length. (Note that if the string contains any literal single quote characters, then these must be represented as a pair of single quote characters in the *FITS* keyword value, and these 2 characters must both be contained within a single substring).
2. Append an ampersand character (‘&’) to the end of each substring, except for the last substring. This character serves as a flag to *FITS* reading software that this string value *may* be continued on the following keyword in the header.
3. Enclose each substring with single quote characters. Non-significant space characters may occur between the ampersand character and the closing quote character.
4. Write the first substring as the value of the specified keyword.
5. Write each subsequent substring, in order, to a series of keywords that all have the name `CONTINUE` in bytes 1 through 8 and have space characters in bytes 9 and 10 of the keyword record. The substring may be located anywhere in bytes 11 through 80 of the keyword record and may be preceded by non-significant space characters starting in byte 11. A comment string may follow the substring; if present, the comment string must be separated from the substring by at least 1 space character followed by a forward slash character (‘/’).

The following keyword records illustrate a string value that is continued over multiple keyword records. (Note: the length of the substrings have been reduced to fit within the page layout):

```
WEATHER = 'Partly cloudy during the evening f&'
CONTINUE  'ollowed by cloudy skies overnight.&'
CONTINUE  ' Low 21C. Winds NNE at 5 to 10 mph.'
```

FITS reading software can reconstruct the long string value by following an inverse procedure of checking if the string value ends with the ‘&’ character and is immediately followed by a conforming `CONTINUE` keyword record. If both conditions are true, then concatenate the substring from the `CONTINUE` record onto the previous substring after first deleting the trailing ‘&’ character. Repeat these steps until all subsequent `CONTINUE` records have been processed.

Note that if a string value ends with the ‘&’ character, but is not immediately followed by a conforming `CONTINUE` keyword, then the ‘&’ character should be interpreted as the literal last character in the string. Also, any ‘orphaned’ `CONTINUE` keyword records should be interpreted as containing commentary text in bytes 9 – 80 (similar to a `COMMENT` keyword).

4,2,2 Logical