Bit flag	Name	Description
0x1	quality scores stored as array	quality scores can be stored as read features or as an array similar to read bases.
0x2	detached	mate information is stored verbatim (e.g. because the pair spans multiple slices or the fields differ to the CRAM computed method)
0x4	has mate downstream	tells if the next segment should be expected further in the stream
0x8	decode sequence as "*"	informs the decoder that the sequence is unknown and that any encoded reference differences are present only to recreate the CIGAR string.

The following pseudocode describes the general process of decoding an entire CRAM record. The sequence data itself is in one of two encoding formats depending on whether the record is aligned (mapped).

Decode pseudocode

```
procedure DecodeRecord
    $BAM\_flags \leftarrow$ ReadItem(BF, Integer)
    $CRAM\_flags \leftarrow$ ReadItem(CF, Integer)
    DecodePositions                                  ▷ See section 10.2
    DecodeNames                                      ▷ See section 10.3
    DecodeMateData                                   ▷ See section 10.4
    DecodeTagData                                    ▷ See section 10.5

    if ($BF$ AND 4) = 0 then                         ▷ Unmapped flag
        DecodeMappedRead                             ▷ See section 10.6
    else
        DecodeUnmappedRead                           ▷ See section 10.7
    end if
end procedure
```

This pseudocode is not meant to be a fully implementable programming language, but to act as an algorithmic guide to the order and structure of CRAM decoding.

The ReadItem function referred above takes two arguments; the data series name and the data type used by the Encoding. It will use the codec specified in the Container Compression Header to retrieve the next value from that data series. Note there is only one permitted data type per data series, so the second argument is redundant and is included only as an aide-mémoire.

10.2 CRAM positional data

Following the bit-wise BAM and CRAM flags, CRAM encodes positional related data including reference, alignment positions and length, and read-group. Positional data is stored for both mapped and unmapped sequences, as unmapped data may still be "placed" at a specific location in the genome (without being aligned). Typically this is done to keep a sequence pair (paired-end or mate-pair sequencing libraries) together when one of the pair aligns and the other does not.

For reads stored in a position-sorted slice, the AP-delta flag in the compression header preservation map should be set and the AP data series will be delta encoded, using the slice alignment-start value as the first position to delta against. Note for multi-reference slices this may mean that the AP series includes negative values, such as when moving from an alignment to the end of one reference sequence to the start of the next or to unmapped unplaced data. When the AP-delta flag is not set the AP data series is stored as a normal integer value.