```
8                                   input[i + 1]);
                } else {
10                          quality = qualityTest(
                input[i - 1], result);
12            }
    } else {
14          quality = qualityTest(
                input[i - 1], result);
16    }
} else if (i > 0 && i == input.length - 1) {
18        quality = qualityTest(input[i - 1],
                                  result);
20
} else {
22        quality = 100;
}
```

This basically calls the getQuality() method, which queries the bigram data and return the frequency. In most cases the parameters are the previous word, then the result of the obfuscation, although for the first word in a sentence the parameters are the result and the next word. If the quality is above a certain value, then the obfuscation is finalised and the appropriate number of elements removed from the bitstream. The input list is updated with the new word, so the quality checks always take into account replacement words.
With deobfuscation, this quality test is performed first. If the result is above the threshold, then it is assumed data is hidden so the synset is generated and deobfuscation is attempted (only if a synset is found). If the result of deobfuscation is not 0, 1 or 2 then the word is ignored. If there is a valid result, then the bits are added to the recovered bits list to be returned to the user.

4.4.2 Synonym Retrieval

For each word in the input, if it is at least 3 characters in length a method, getFullSet() is called which assembles the synset using the getWord() function. This function queries the MasterIndex table, as described above, and then calls four individual classes for fetching each of the noun, verb, adverb and adjective senses (only if there are any listed in the index file). These classes simply perform a query on their appropriate data table and return the synonyms as a two-dimensional ArrayList of String objects. These are stored in a Word object, which is used by the getFullSet method to build the initial synset. It then repeats this process for each of the words in the synset, and adds their synonyms to the set, ignoring duplications. The quality of each of these words is found using the getQuality method (using the word before and the current word as parameters), the result of which is stored, along with