In a spelling checker dictionary, tokenising is used at the front of words. Consider, for example, the large collection of words that begin with auto-, non-, dis- or con-. The VizaSpell package, which runs with the VizaWrite word processor on the Commodore 64, makes use of both compression and tokenising to squeeze a 30,000 word dictionary into a mere 65 Kbytes on disk.

The most dificult task of a spelling checker, however, is to look up all the words from a document in its dictionary. A binary search could be used (see page 416), but for a thousand word document this could take hours. Ideally, the word processor should check each word as it is typed, but this is impractical in programming terms and therefore a document will usually be checked as a whole, either on disk or (on larger machines) in RAM. The program works through the document and compiles a list of the words it contains in alphabetical order. It is not unusual for more than 50 per cent of a large report to be made up from just 100 different words.

Most spelling checkers use this process to provide a useful additional report on the usage of words in your document — which may help you to spot unnecessary repetition. A simple algorithm

then works its way through this list and the dictionary list simultaneously, looking for matches. In this way, the time taken to complete the search will be greatly reduced and constant — four minutes in the case of VizaSpell, irrespective of the document's length.

Words that are not found in the dictionary will either be printed out as a list, or highlighted within the original document. For each highlighted word, the user is presented with three options:

1) The word has been mis-spelled or mis-typed and should be corrected;
2) The word is correct and should be added to the program's dictionary;
3) The word is correct, but is unlikely to be used again (e.g. it is part of an address), so it should be left alone, and not added to the dictionary.

Grammar and style checkers work in a similar manner. The former work on a limited number of rules (such as looking for a capital letter at the start of every sentence) and, consequently, there are many grammatical inaccuracies that won't be picked up. Style checkers are still in their infancy, and most of the packages currently available simply make use of a large dictionary of examples in order to identify bad syntax and expressions. Generally, these packages will suggest better ways of phrasing the clumsy constructions that they find, by referring to their dictionaries. They will also usually pick up an excessive use of a word phrase within a paragraph, or the use of long and inelegant sentences.

POPPERFOTO

IAN McKINNELL

To Be Or Not To Be
Think how much easier English Literature would be if spelling checker programs were allowed into the exam room! We can use Hamlet's soliloquy to illustrate how one such program (VizaSpell) works. First, the text is typed into the computer using a word processor. Then the spelling checker is invoked with a couple of simple commands, and this creates an alphabetic list of all the words used, also indicating their frequency of use. This list is checked against the dictionary on disk, and unrecognised words are highlighted. When first used, the program may highlight some seemingly common words, but these can be added to the dictionary for later use

Writing a simple form of spelling, grammar or style checker in BASIC can be a very interesting exercise even for an inexperienced programmer — though you will need a fairly good knowledge of the string-handling functions on your machine. As software sophistication increases, it would seem reasonable to expect word processing packages to come with such functions built-in, and more as well. Ah yes, what every writer would adore: 'COMMAND > GENERATE ARTICLE, LENGTH 1200 WORDS, BEGIN'