Sentinel Loop Patterns

How do you handle problems where the loop <mark>reads data from the user</mark> until some special value, or **sentinel**, is found to signal the end of the input? This is called a <mark>sentinel loop</mark> and its logical structure is:

```
Read a value
If the value is equal to the sentinel then
    Exit the loop
Process the value
```

There is no easy test at the beginning of the loop; you don't know when the sentinel is encountered <mark>until you've read a value</mark>. There are three ways to **rearrange** the statements to handle this situation.