When \( i \) reaches the value 102, the loop quits. The main point here is that the update expression can be any valid expression. For example, if you want to square \( i \) and add 10 each cycle, you can use \( i = i * i + 10 \).

### Inside Strings with the for Loop

The **for** loop provides a direct way to access each character in a string in turn. **Listing 5.6**, for example, enables you to enter a string and then displays the string character-by-character in reverse order. The `strlen()` yields the number of characters in the string; the loop uses that value in its initializing expression to set \( i \) to the index of the last character in the string, not counting the null character. To count backward, the program uses the decrement operator (\(--\)) to decrease the array subscript by 1 each loop. Also, **Listing 5.6** uses the greater-than or equal-to relational operator (\( \geq \)) to test whether the loop has reached the first element. We'll summarize all the relational operators soon.

**Listing 5.6 forstr1.cpp**

```cpp
// forstr1.cpp -- using for with a string
#include <iostream>
#include <cstring>
using namespace std;
const int ArSize = 20;
int main()
{
    cout << "Enter a word: ";
    char word[ArSize];
    cin >> word;

    // display letters in reverse order
    for (int i = strlen(word) - 1; i >= 0; i--)
        cout << word[i];
    cout << "\nBye.\n";
    return 0;
}
```