Start as Before, with Command-Line Arguments

Let's take a look at the code.

```
static const
    int32_t max_word_len = 500;
int main (int argc, char* argv[])
{
    char  buf1[max_word_len + 1];
    char  buf2[max_word_len + 1];
    char* last_line;
    char* cur_line;
```

limit on word size

command-line arguments

41

Declare Buffers and Pointers to Buffers

```
static const
    int32_t max_word_len = 500;
int main (int argc, char* argv[])
{
    char  buf1[max_word_len + 1];
    char  buf2[max_word_len + 1];
    char* last_line;
    char* cur_line;
```

include space for NUL

pointers for double-buffering

42

Need Variables for Swapping and Counting

```
char*   tmp;
int32_t count;
if (1 != argc) {
    fprintf (stderr,
        "syntax: %s\n",
        argv[0]);
    return EXIT_BAD_ARGS;
}
```

Swapping requires a temporary.

count of current word

43

Start by Checking Command Syntax

```
char*   tmp;
int32_t count;
if (1 != argc) {
    fprintf (stderr,
        "syntax: %s\n",
        argv[0]);
    return EXIT_BAD_ARGS;
}
```

must be called with one argument

error messages sent to **stderr**

the program's name

Lumetta's favorite return value for bad arguments (enumerated, not 0)

44