

Try to Open the File

```

if (NULL == (in_file =
    fopen (argv[1], "r"))) {
    perror ("open file");
    return EXIT_FAIL;
}

write = buf;
word_len = 0;

```

Try to open file.

Print error message on `fopen` failure.

End the program, returning another enumerated constant.

21

Initialize the FSM

```

if (NULL == (in_file =
    fopen (argv[1], "r"))) {
    perror ("open file");
    return EXIT_FAIL;
}

write = buf;
word_len = 0;

```

Initialize the Finite State Machine.

22

Execute FSM, then Handle Any Final Word

```

while (EOF !=
    (a_char = getc (in_file))) {
    // execute FSM
}

if (0 < word_len) {
    *write = 0;
    puts (buf);
}

(void)fclose (in_file);
return EXIT_SUCCEED;

```

Process input stream one character at a time.

Did file end with a word left in the buffer?

If so, write word out (with a newline).

23

Close the Stream and Return Success

```

while (EOF !=
    (a_char = getc (in_file))) {
    // execute FSM
}

if (0 < word_len) {
    *write = 0;
    puts (buf);
}

(void)fclose (in_file);
return EXIT_SUCCEED;

```

Close the input stream.

Return 0 to indicate successful execution.

24