# Count lines of input

```
#include <stdio.h>

void main(void)
{
    int c;
    int numberOfLines = 0;
    while ((c = getchar()) != EOF)
    {
        if(c == '
') numberOfLines ++;
        if(c == '
')
    }
    printf("%d\n", numberOfLines);
}
```
```c
#include <stdio.h>

void main(void)
{
    int numberOfLines = 0;
    int c = getchar();

    while (c != EOF)
    {
        if(c == '\n') numberOfLines ++;
        c = getchar();
    }

    printf("%d\n", numberOfLines);
}
```

Why is `getchar()` called on two different lines of code?

Compile: `gcc lineCounter.c`
Run with input: `./a.out < lineCounter.c`
Output: 13
#include <stdio.h>
#define IN 1
#define OUT 0

void main(void)
{
    int c, nl, nw, nc, state;
    state = OUT;
    nl = nw = nc = 0;
    while((c = getchar()) != EOF) {
        ++nc;
        if (c == '\n')
            ++nl;
        if(c == ' ' || c == '\n' || c == '\t')
            state = OUT;
        else if(state == OUT) {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
}

Coding style in
textbook:

- Variable names
too short.
- Multiple actions
  in one line.
- Leaves out {} when body of block is only one statement.
- { at end of line.
#include <stdio.h>

#define IN 1
#define OUT 0

void main(void)
{
    /* Body of function on next slides */
}
Counting it all – Main

```c
void main (void)
{
    int charCount = 0;
    int totalCharCount = 0;
    int wordCount = 0;
    int totalWordCount = 0;
    int wordState = OUT;
    int lineCount = 1;
    int c = getchar();

    while (c != EOF)
    {
        /* BODY ON NEXT SLIDE */
    }

    printf("%d lines, %d words, %d characters\n", lineCount -1, totalWordCount, totalCharCount);
}
```
if (charCount == 0)
{
    printf("%2d. ", lineCount);
}

if (c == '
')
{
    printf("(%d;%d)\n", charCount, wordCount);
    charCount = 0;
    wordCount = 0;
    wordState = OUT;
    lineCount++;
}
else
else /* char just read not '\n' */
{
    charCount ++;
    totalCharCount ++;
    printf("%c", c);
    if (c == ' ' || c == '\n' || c == '\t')
    {
        wordState = OUT;
    }
    else if (wordState == OUT)
    {
        wordState = IN;
        wordCount ++;
        totalWordCount ++;
    }
}

c = getchar();
Finding longest lines

Add some variables for char counts and line numbers.

```plaintext
int mostCharLineNum = 0;
int mostCharCount = 0;
int mostWordLineNum = 0;
int mostWordCount = 0;
```

Output results at end

```plaintext
printf("%d lines, %d words, %d characters\n", lineCount-1, totalWordCount, totalCharCount);
printf("Line %d has the most words with %d\n", mostWordLineNum, mostWordCount);
printf("Line %d has the most characters with %d\n", mostCharLineNum, mostCharCount);
```
Finding longest lines

At end of lines, check for new longest line.

```c
if (c == '\n')
{
    printf("(%d;%d)\n", wordCount, charCount);

    if(charCount >= mostCharCount)
    {
        mostCharCount = charCount;
        mostCharLineNum = lineCount;
    }

    if(wordCount >= mostWordCount)
    {
        mostWordCount = wordCount;
        mostWordLineNum = lineCount;
    }

    charCount = 0;
    wordCount = 0;
    wordState = OUT;
    lineCount++;
}
else
```