Watch out for other restrictions on identifiers. Some compilers treat certain identifiers (`asm`, for example) as additional keywords. Identifiers that belong to the standard library are restricted as well. Accidentally using one of these names can cause an error during compilation or linking. Identifiers that begin with an underscore are also restricted.

restrictions on identifiers ➤21.1

## 2.8 Layout of a C Program

We can think of a C program as a series of *tokens:* groups of characters that can't be split up without changing their meaning. Identifiers and keywords are tokens. So are operators like + and -, punctuation marks such as the comma and semicolon, and string literals. For example, the statement

```
printf("Height: %d\n", height);
```

consists of seven tokens:

```
printf    (    "Height: %d\n"    ,    height    )    ;
  ①       ②          ③          ④       ⑤       ⑥    ⑦
```

Tokens ① and ⑤ are identifiers, token ③ is a string literal, and tokens ②, ④, ⑥, and ⑦ are punctuation.

The amount of space between tokens in a program isn't critical in most cases. At one extreme, tokens can be crammed together with no space between them at all, except where this would cause two tokens to merge into a third token. For example, we could delete most of the space in the `celsius.c` program of Section 2.6, provided that we leave space between tokens such as `int` and `main` and between `float` and `fahrenheit`:

```
/* Converts a Fahrenheit temperature to Celsius */
#include <stdio.h>
#define FREEZING_PT 32.0f
#define SCALE_FACTOR (5.0f/9.0f)
int main(void){float fahrenheit,celsius;printf(
"Enter Fahrenheit temperature: ");scanf("%f", &fahrenheit);
celsius=(fahrenheit-FREEZING_PT)*SCALE_FACTOR;
printf("Celsius equivalent: %.1f\n", celsius);return 0;}
```

In fact, if the page were wider, we could put the entire `main` function on a single line. We can't put the whole *program* on one line, though, because each preprocessing directive requires a separate line.

Compressing programs in this fashion isn't a good idea. In fact, adding spaces and blank lines to a program can make it easier to read and understand. Fortunately,