Magic Numbers

In their famous book, "The C Programming Language", Kernighan and Ritchie (the inventors of C) spend some time discussing a C program similar to the C++ program below. The program prints a table of Fahrenheit temperatures and their Celsius equivalents.

```
main()
{
  int Fahrenheit;

  Fahrenheit = 0;
  while (Fahrenheit <= 300)
  {
    Fahrenheit = Fahrenheit + 20;
    cout << endl << setw(4) << Fahrenheit << setprecision(2)
         << setw(8) << (5.0/9.0 * (Fahrenheit – 32));
  }
}
```

K & R make the following statement.

> It's bad practice to bury "magic numbers" like `300` and `20` in a program; they convey little information to someone who might have to read the program later, and are hard to change in a systematic way. One way to deal with magic numbers is to give them meaningful names…

K & R then rewrite the program, using **define** statements, but constants could also be use, to replace the `0` with "`LOWERLIMIT`", the `300` with "`UPPERLIMIT`" and the `20` with "`STEP`".

Good programmers agree that most numbers or literals used in programs should be given names. The most important reasons were noted in the K & R statement above:

1) If constants are given descriptive names that reflect the purpose of the numbers, the program will be easier to understand.

2) Programs that use named constants are much easier to modify.

Names for constants should be chosen carefully and should reflect the use of the constant or number. In the program above, for example, it would be a mistake to use "`ZERO`" instead of "`LOWERLIMIT`".

A convention for naming constants that is used frequently is that the name of the constant be all upper case letters.

Note that changing "5.0/9.0" to a named constant will probably not make the program above easier to read. Also, the presence of this fraction will not make the program harder to modify, since it will never have to be changed. This suggest the following informal rule:

> A number or literal constant should probably not be named if naming it will not make the program easier to read and the value of the number or literal constant will never change.