

## The Digit Separator (')

A digit separator is a single-character token (') that can appear as part of a numeric literal without altering its value.

### Description

A digit separator — i.e., an instance of the single-quote character (') — may be placed anywhere within a numeric literal to visually separate its digits without affecting its value:

```
int      i = -12'345;           // same as -12345
unsigned int u = 1'000'000u;    // same as 1000000u
long     j = 500'000L;         // same as 500000L
long long k = 9'223'372'036'854'775'807; // same as 9223372036854775807
float    f = 3.14159'26535f;    // same as 3.1415926535f
double   d = 3.14159'26535'89793; // same as 3.141592653589793
long double e = 20'812.80745'23204; // same as 20812.8074523204
int      hex = 0x0C'25'00'F9;    // same as 0x0C2500F9
int      oct = 044'73'26;        // same as 0447326
int      bin = 0b1001'0110'1010'0111; // same as 0b1001011010100111
```

Multiple digit separators within a single literal are allowed, but they cannot be contiguous, nor can they appear either before or after the *numeric* part, i.e., digit sequence, of the literal:

```
int e0 = 10''00; // Error, consecutive digit separators
int e1 = -'1000; // Error, before numeric part
int e2 = 1000'u; // Error, after numeric part
int e3 = 0x'abc; // Error, before numeric part
int e4 = 0'xdef; // Error, way before numeric part
int e5 = 0'89;   // Error, nonoctal digits
int e6 = 0'67;   // OK, valid octal literal
```

Although the leading `0x` and `0b` prefixes for hexadecimal and binary literals, respectively, are not considered part of the *numeric* part of the literal, a leading `0` in an octal literal is. As a side note, remember that on some platforms an integer literal that is too large to fit in a **long long int** but that does fit in an **unsigned long long int** might generate a warning or error<sup>1</sup>:

```
unsigned long long big1 = 9'223'372'036'854'775'808; // 2^63
// warning: integer constant is so large that it is an
// unsigned long long big1 = 9'223'372'036'854'775'808;
//                               ^-----
```

<sup>1</sup>Tested on GCC 7.4.0 (c. 2018).