The strcmp Function

Like `strcpy()`, most implementations of `strcmp()` are cryptic. Here's the version from GNU C:

```
int strcmp(const char *s1, const char *s2)
{
    const unsigned char *a1, *a2;
    for (a1 = reinterpret_cast<const unsigned char *>(s1),
         a2 = reinterpret_cast<const unsigned char *>(s2);
         *a1 == *a2; a1++, a2++)
        if (*a1 == '\0') return 0;
    return *a1 - *a2;
}
```

The GNU version of `strcmp()` returns **the difference** between the first two mismatched characters. `a1` and `a2` are temporary pointers to `unsigned char`, so the characters can be interpreted as raw values between `0-255`. The pointers are initializaed by using a `reinterpret_cast`.

Here is an alternate (Apple/Next/PPC) version of the same function, which returns `0`, `+1` and `-1` instead of the difference between the characters. This version, written in 1992, uses traditional C-style casts to handle the `signed`/`unsigned` instead of a C++ `reinterpret_cast`.

```
int strcmp(const char *s1, const char *s2)
{
    for ( ; *s1 == *s2; s1++, s2++)
        if (*s1 = '\0') return 0;   // reached both NULs. Equal
    return ((*(unsigned char *)s1
            < *(unsigned char *)s2) ? -1 : +1);
}
```