`sizeof()` operator

If you have any doubts as to the amount of memory actually used by any variable (and this goes for types we'll discuss later, also), you can use the `sizeof` operator to find out for sure. (For completeness, it is important to mention that `sizeof` is a unary operator, not a function.) Its syntax is:

```
sizeof object
sizeof(type)
```

The two expressions above return the size of the object and type specified, in bytes. The return type is `size_t` (defined in the header `<stddef.h>`) which is an unsigned value. Here's an example usage:

```
size_t size;
int i;
size = sizeof(i);
```

`size` will be set to 4, assuming `CHAR_BIT` is defined as 8, and an integer is 32 bits wide. The value of `sizeof`'s result is the number of bytes.

Note that when `sizeof` is applied to a `char`, the result is 1; that is:

```
sizeof(char)
```

always returns 1.