【C】Day1(2)

Course	Advanced C
Study Date	@March 15, 2022

【Ch6】Struct

6.3 Arrays of Structures

Consider writing a program to count the occurrences of each C keyword.

We need an array of character strings to hold the names, and an array of integers for the counts. One possibility is to use two parallel arrays, `keyword` and `keycount`, as in:

```
char *keywoard[NKEYS];
int keycount[NKEYS];
```

But we can use an array of structure, each keyword entry is a pair:

```
struct key {
  char *word;
  int count;
} keytab[NKEYS];
```

The structure declaration declares a structure type `key`, defines an array `keytab` of structure of this type, and sets aside storage for them. Each element of the array is a structure.

Since the structure `keytab` contains a constant set of names, it is easiest to make it an external variable and initialize it once and for all when it is defined.

```
struct key {
  char *word;
  int count;
} keytab[] = {
  "auto", 0,
  "break", 0,
  "case", 0,
  "char", 0,
  "const", 0,
  "continue", 0,
  "default", 0,
  ...
};
```

It would be more precise to enclose initializers for each structure in braces:

```
{ "auto", 0 },
{ "break", 0 },
{ "case", 0 },
```

The number of entries in the array `keytab` will be computed if initializers are present and the `[]` is left empty.