1. The next few sections contain stuff from the file `"lcommon.w"` that must be included in both `"lit.w"` and `"log.w"`. It appears in file `"lcommon.h"`, which is also included in `"lcommon.w"` to propagate possible changes from this `COMMON` interface consistently.

2. Include the printf declaration.

```
#include <stdio.h>
```

3. Declaration of the global variables or function simply declares that the variable or function exists, but the memory is not allocated for them.

```
  extern int argc;     /* copy of ac parameter to main */
  extern char **argv;     /* copy of av parameter to main */
```

4. `LIT` has a fairly straightforward outline. It operates in three phases: First it inputs the source file and stores cross-reference data, then it inputs the source once again and produces the TEX output file, finally it sorts and outputs the index.

5. *main()* - is called by the C library by recognizing the in-built keyword *main*. The way for running another program on `Linux` involves first calling *fork()*, which creates a new process as a copy of the first one, and then calling *exec()* to replace this copy (of the shell) with the actual program to run.

Richtie and Kernighan write: "... *main* is a special function. Our program begins executing at the beginning of *main*. This means that every program must have a *main* somewhere and will usually call other functions to help perform its job."

ptr:	the *&struct list_head* pointer.
type:	the type of the struct this is embedded in.
member:	the name of the list_head within the struct.

return xxx.

```
  void main(void)
  {
    printf("Hi␣Herbert␣and␣Renate.\n");
  }
```

6. Index.

LIT