

文件系统

现有一个新设计的文件系统，它是由多种类型的文件构成的，每个文件包含了文件名、文件类型、所属用户、所属群组、文件大小这些基本属性。（限定文件名在系统中不会重复）

文件类型有：

- 普通文件，类型代号为c，包含了文本文件以及二进制文件。
- 目录文件，类型代号为d，目录文件下能够存放其他类型的文件。
- 镜像文件，类型代号为l，可以把它看成是对源文件的一个拷贝。不过如果删除了镜像文件，源文件也会被一并删除；删除源文件则不会影响镜像文件。另外，镜像文件的所属用户、所属群组和文件大小和源文件相同。

定义一系列的命令来操作文件 ({}中的内容均不含空格)：

- 新建文件：（如果目标目录文件名为null，则表示在根目录中添加文件）
 - 新建普通文件：`touch {目标目录文件名} {文件名} {所属用户} {所属群组} {文件大小}`，表示向目标目录中添加一个新的普通文件，如 `touch bd 1.txt stu cpp 12` 表示向目录文件bd中添加一个名为1.txt的普通文件，文件大小为12，文件所属的用户和群组分别为stu和cpp。
 - 新建目录文件：`mkdir {目标目录文件名} {文件名} {所属用户} {所属群组}`，表示向目标目录中添加一个新的目录文件，如 `mkdir bd ac stu cpp` 表示向目录文件bd中添加一个名为ac的目录文件，文件所属的用户和群组分别为stu和cpp。
 - 新建镜像文件：`ln {目标目录文件名} {新建的文件名} {源文件名}`，表示向目标目录中添加一个新的镜像文件，如 `ln bd link 1.txt` 表示在目录文件bd下对1.txt创建一个名为link的镜像文件，link文件的所属用户、所属群组和文件大小和1.txt相同。
- 删除文件：`rm {文件名}`，如果是普通文件，直接执行删除；如果是目录文件，需要递归删除掉这个目录下的所有文件；如果是镜像文件，需要删除该文件和它的源文件（如果它的源文件也是镜像文件，需要一直删除下去，直到最终的源文件被删除）。
- 根据文件名查询文件：`ls {文件名}`，按照 `{文件名} {文件类型} {所属用户} {所属群组} {文件大小}` 输出该文件的属性，文件类型输出类型的相应代号即可。
 - 如果是普通文件和镜像文件，按照此格式输出。
 - 如果是目录文件，则先按照格式输出当前这个目录文件，然后再输出目录下所有的文件（若这些文件中也有目录文件，则需要输出这个子目录下的子文件，即深度优先），对于处在同一目录的多个文件，按照它们被加入到文件的顺序输出。（目录文件的大小是它子文件的大小总和。）

示例：

输入：

```
mkdir null bd stu cpp
touch null a.java stu cpp 12
ls a.java
touch bd b.txt stu cpp 1
mkdir bd ac stu cpp
touch bd a.cc stu cpp 1
touch ac b.md stu cpp 1
ls bd
```

输出:

```
a.java c stu cpp 12
bd d stu cpp 3
b.txt c stu cpp 1
ac d stu cpp 1
b.md c stu cpp 1
a.cc c stu cpp 1
```