Problem 5: Subdirectories (3 pages)

Your program will simulate the creation of subdirectories (folders) on one of the disks of a computer. The input file to your program, prog5.dat, will contain a sequence of commands that a user might enter from a command line, and the output file prog5.out will contain the operating system's responses to these commands. Below is an example of an input file, and on the right is the listing of the corresponding output file.

```
dir
mkdir   sub6
mkdir   sub3
mkdir   sub4
dir
mkdir   sub4
cd      sub3
cd      sub3
mkdir   sub3
mkdir   sub6
mkdir   sub4
dir
cd      sub6
mkdir   sub666
dir
up
up
dir
up
```

```
Problem 5 by team X
Command: dir
Directory of root:
No subdirectories
Command: mkdir   sub6
Command: mkdir   sub3
Command: mkdir   sub4
Command: dir
Directory of root:
sub3    sub4    sub6
Command: mkdir   sub4
Subdirectory already exists
Command: cd      sub3
Command: cd      sub3
Subdirectory does not exist
Command: mkdir   sub3
Command: mkdir   sub6
Command: mkdir   sub4
Command: dir
Directory of root\sub3:
sub3    sub4    sub6
Command: cd      sub6
Command: mkdir   sub666
Command: dir
Directory of root\sub3\sub6:
sub666
Command: up
Command: up
Command: dir
Directory of root:
sub3    sub4    sub6
Command: up
Cannot move up from root directory
End of problem 5 by team X
```

The four commands that can appear in the input file are:

`dir`		Display the path and the subdirectories of the current default directory, the latter in lexicographic order.
`mkdir`	`<name>`	Create a subdirectory of the current default directory with the specified name.
`cd`	`<name>`	Change the default to a specified subdirectory of the current default directory.
`up`		Change the default to the parent directory of the current default directory.

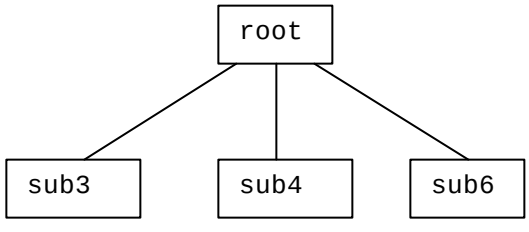
- Each line of the input file begins with one of the four commands, in lower case letters.
- The commands `mkdir` and `cd` in the input file will be followed (starting in column 9) by an argument (the subdirectory name).
  - The argument will consist of at least one and at most six characters that can be upper or lower case letters, digits, or underscores; in particular, the argument will not contain blanks.
  - The argument will be followed immediately by the end-of-line character.
  - The commands `dir` and `up` do not take an argument, they will be followed immediately by the end-of-line character.

- Before responding to each command in the input file, your program will echo the command :
  - The command will be displayed starting in column 10 (after "`Command: `");
  - If there is an argument, it will be displayed starting in column 18.
- In response to the `dir` command, your program will:
  - Display the path from the root directory to the current default directory, in the format of the sample output
    - In the example shown, the `dir` command appears four times in the input file. The first two times the path is `root`, the second time it is `root\sub3`, the third time it is `root\sub3\sub6`.
  - If the current default directory contains no subdirectories, display the message to that effect; otherwise display in lexicographic order all subdirectories of the current default directory.
    - The subdirectories will be left-justified in fields of width 8.
    - There is no specific limit on the number of subdirectories of a particular directory. If there are more than ten, the displayed list will wrap around to the beginning of the next line after each group of ten subdirectories. For example, if the subdirectory `sub666` in the above example had 20 siblings `sub601`, `sub602`, etc., the output from the `dir` command would be:

```
Directory of root\sub3\sub6:
sub601  sub602  sub603  sub604  sub605  sub606  sub607  sub608  sub609 sub610
sub611  sub612  sub613  sub614  sub615  sub616  sub617  sub618  sub619 sub620
sub666
```

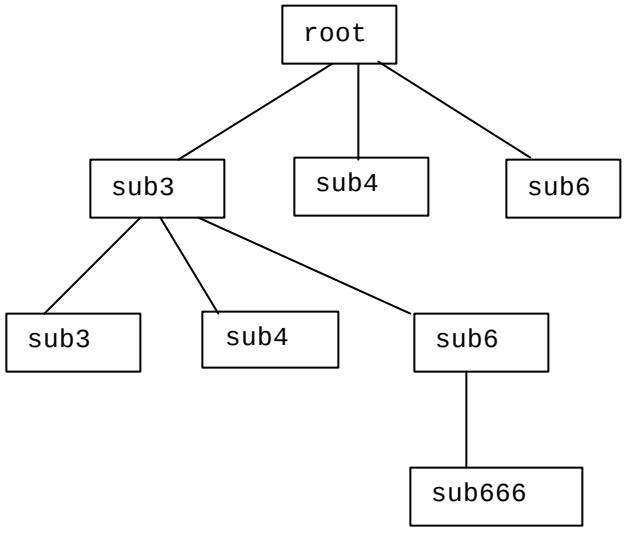
- The commands `mkdir`, `cd`, and `up` will not produce any immediate response, unless an error message is in order. The reasons for possible error messages are as follows:
  - In response to the command `mkdir`: the current default directory already has a subdirectory whose name is specified in the argument.
  - In response to the command `cd`: the current default directory does not have a subdirectory whose name is specified in the argument.
  - In response to the command `up`: the current default directory is the root directory.
- At program startup, the default directory is the root directory, and it has no subdirectories.

In the example shown above, the effect of the first group of three `mkdir` commands is to create the directory tree shown on the right.



After changing the default directory to sub3, executing the second group of three `mkdir` commands, changing the default directory to the subdirectory `sub6` and executing the last `mkdir` command, the resulting directory tree is shown on the right.

Note that subdirectories that are not siblings (do not have the same parent) may have the same name.



✍✍There is no limit on the number of subdirectories of a particular directory, nor is there a limit on the maximum number of levels in the directory tree. The total number of subdirectories successfully created by the program will not, however, exceed 5000.

✍✍Pay close attention to every detail of the output, such as wording and punctuation, upper/lower case variations, number of blank spaces, and the absence of blank lines.

✍✍A few lines of the above output are reproduced here with a formatting template:

```
12345678901234567890123456789O
Command: mkdir   sub4
Command: dir
Directory of root\sub3:
sub3    sub4    sub6
Command: cd      sub6
Command: mkdir   sub666
```