

4427 deltree

You have just run out of disk space and decided to delete some of your directories. Rationally, you will first have an exploration of what you have in your file system. And more rationally, you will do this exploration through a command line interface. The interface used in this problem is called “MSDOS--”, since it is something like MSDOS with fewer features. The commands of MSDOS-- are as follows:

- `cd < directory >`

Assuming `< directory >` to be the name of a relative descendant of **current directory**, this command changes the current directory to `< directory >`. For example, when the **current directory** is “\A\B\” and one of its descendants is “C\D”, the execution of “`cd C\D`” will change the **current directory** to “\A\B\C\D”.

- `cd \`

This command changes the **current directory** to “\” (the root of the file system). For example, when the current directory is “\A\B\”, the execution of “`cd \`” will change the **current directory** to “\”.

- `cd ..`

Assuming the **current directory** to be anything except “\”, this command changes **the current directory** to its parent directory. For example, when the current directory is “\A\B\”, the execution of “`cd ..`” will change the current directory to “\A\”.

- `cd \< directory >`

This command is equivalent to the execution of the following two commands:

```
cd \  
cd < directory >
```

- `dir`

This command lists the name of files and directories directly in the **current directory**, each on a separate line. These file/directory names are made up of (lowercase and uppercase) letters, digits, and dots (“.”). Directory names precede the file names in the list, and each one, comes alone in a single line. On the contrary, each file name is accompanied by its size separated by a space. A sample output of “`dir`” is as follows:

```
HW1  
HW1.old  
Syllab.pdf 10000  
notes.txt 3241
```

- `deltree < directory >`

Assuming `< directory >` to be the name of a relative descendant of **current directory**, this command tries to delete `< directory >` and all its descendant files and subdirectories (and thus, freeing that much of space). For example, when the **current directory** is “\A\B\” and one of its descendants is “C\D”, the execution of “`deltree C\D`” will try to delete directory “\A\B\C\D” and all of its descendant files and directories.

- `deltree \< directory >`

This command is equivalent to the execution of the following two commands:

```
cd \
```

```
deltree < directory >
```

- `exit`

This command terminates the command line interface.

A “scenario” is an exploration (a consistent series of “`cd`” and “`dir`” commands and their results, starting from root) followed by exactly one “`deltree`” command. Given a scenario, you are to find the maximum space guaranteed to be freed by executing its “`deltree`” command.

Input

Input contains multiple independent scenarios. There is an empty line after each scenario. The input ends with an ‘`exit`’ command. There is a ‘`>`’ sign before each command in the input (with no spaces in between). The length of each file name does not exceed 50. You may assume that the input is correct.

Output

Write the result of the i -th scenario as a single integer on the i -th line of output.

Sample Input

```
>cd A
>dir
B
C
d 12
e 62
>cd B
>cd ..
>cd ..
>deltree A
```

```
>dir
G
s 2
>cd G
>dir
>cd \
>deltree G
```

```
>dir
A
B
x 3
>cd A
>dir
AA
AB
```

```
ax 10
ay 12
>cd AA
>dir
d 32
a 28
>cd ..
>cd AB
>dir
F
x 100
>cd F
>dir
G
>cd \
>deltree A

>cd D1\D2
>dir
D3
a 32
>cd D3
>dir
b 31
>cd \D1\D3
>dir
d 7
>deltree \D1

>exit
```

Sample Output

```
74
0
182
70
```