

Lab2 - A

Edit

File_System

Description

The sample Input/Output has been updated

This task is to implement some simple UNIX programs. There are totally n create or update command, and m query command. **This task only involves two file types: file and folder.**

Initially, there are no files or folders. You can consider you step into an empty folder, and open the terminal there.

Below is a detailed introduction to all the commands involved in this task:

Create/Update Command

```
echo [content] > [path]
```

This means create a **file** with content `[content]` on `[path]`. The `[content]` only consists of letters ($a-z, A-Z$) and 0-9 digits, or `[content]` is empty and just create a **file** on `[path]`. The path could contain `.` and `..` in this task for all commands. `.` denotes the current directory, and `..` denotes the previous directory. Notice that there won't be the data go to the previous directory for the initial directory. All the path is valid.

```
mkdir [path_to_folder]
```

Create a **folder** on `[path_to_folder]`

eg: `mkdir a`. This will make a folder on initial directory, and `mkdir a/b` will create a folder under folder `a`.

```
rm [path_to_file]
```

remove the **file** on `[path_to_file]`.

```
rm -rf [path_to_file]
```

Recursively delete **files** and **folders** of `[path_to_file]` without giving any warning or confirmation.

```
mv [src_path_to_file] [dst_path_to_file]
```

move **file** or **folder** from `[src_path_to_file]` to `[dst_path_to_file]`. The `[dst_path_to_file]` is a path to folder.

eg: `mv a b`. And this will make `a` under `b`, that is `b/a`.

Query Command

```
cat [path_to_file]
```

Check the content of the **file** on `[path_to_file]`.

```
find [path] [expression]
```

Find all the path from `[path]` and satisfy the `[expression]`.

`[expression]` consists two parts:

`-type f/d` the file type is file or folder.

`-name [name]` the file name equals to `[name]`.

The `[path]` here can be empty, that means begin finding on initial directory `.`

`[expression]` can be empty, one or two.

If both arguments are empty, just output all the paths of file and folder, including the initial directory `.`

For each `cat` command, just output the content, and for `find` command, output the total number and all the paths.

All the commands are valid, and all the file names are different in the same directory(include the initial directory).

Input format

The first line two integers n, m .

The following n lines denote the create or update command.

Then following m lines denote the query command.

Output format

For each `cat` command, one line to output the content.

For each `find` command, first line output the total number of paths. Then following the number of lines, each line output the path, you can output the paths on arbitrary order.

Sample

Sample Input

```
16 6
mkdir a
echo 123 > a/b
echo 234 > a/c
echo 345 > a/d
mkdir b
mkdir c
mkdir d
echo 666 > b/a
```

```
echo 23333 > c/a
echo 12312dasdasdf > d/a
mkdir a/e
echo > a/e/b
echo > b/b
echo > d/b
rm a/c
rm -rf a/e
cat d/a
cat c/a
cat d/b
find
find a -name b
find ../././ -name b -type f
```

Sample Output

```
12312dasdasdf
23333

12
.
./a
./a/b
./a/d
./b
./b/a
./b/b
./c
./c/a
./d
./d/a
./d/b
1
a/b
3
../././a/b
../././b/b
../././d/b
```

You can use your own UNIX system to run the command to check the result.

Constraints:

For 100% testcases, $1 \leq n \leq 550, 1 \leq m \leq 10$. The length of content and path will not exceed 50.

For 30% testcases, there are only `echo` operations for the n commands.

For 60% testcases, there are not update commands.

Lab2 - B

Edit

Counting

Description

Given an undirected connected graph G with n nodes and m edges. Nodes are numbered starting from 1 to n .

Given two integers a, b . Now counting the pairs (x, y) that any path from node x to node y goes through node a and node b ($x \neq a, x \neq b, y \neq a, y \neq b$).

Print the required number of pairs. The order of two nodes in a pair does not matter, that is, the pairs (x, y) and (y, x) must be taken into account only once.

Input Format

The first line contains four integers, n, m, a, b .

Then following m lines, each line contains two integer u, v denoting that there is an undirected edge between node u and node v .

Output Format

One line the total number of required pairs.

Sample

Sample Input1

```
7 7 3 5
5 6
6 7
7 5
1 2
2 3
3 4
4 5
```

Sample Output1

```
4
```

Sample Input2

```
6 6 2 3
1 2
2 3
3 4
4 6
5 6
1 5
```

Sample Output2

0

Constraints

For 50% testcases: $m=n-1$.

For 100% testcases: $4 \leq n \leq 1 \times 10^5, n-1 \leq m \leq 2 \times 10^5, a, b \in [1, n]$.