Write a program which receives on the command line **two arguments**, intended to be the names of two files containing:

- **First file**:
  - A list of strings (tokens), composed only by alfanumeric characters [a-z,A-Z,0-9], no blanks.
  - One token per line, lines are terminated by '\n' (newline).
  - Max length of a line: 80 chars.
- **Second file**:
  - A list of file names (paths), one path per line.
  - Max length of a line: 80 chars.

The program will then:

- Read and parse the two files **only once**.
- Look for each token, found in the first file, in all the files (path names) indicated in the second file. You can assume that lines contained in these files are long, at most, 80 characters.
- Output the statistics by printing, for each token, the files (names in the second file) where tokens where found and the number of occurrences (in each file).
- Note: You can search only for "full line" matches (e.g. *abc* does **not** match *abcd* or *d abc*, etc.).

Hints:

- Start designing the modules, not writing code.
- Use library functions (strings, characters..), don't reinvent the wheel.
- Don't spend time (initially) chasing perfection. E.g. skip error checking, optimizing memory allocation, etc, but instead focus on the algorithm.
- Send me the code even if not completed/compilable/running, it has a value.

Assuming :

FileA

```
aa
bbbb
c
d1234
```

FileB

```
pippo.txt
pluto.txt
```

pippo.txt

```
sadas
bbbb
c
bbbb
abcd123
```

pluto.txt

```
d1234
c
bbbbb
zz987
```

Then (`a.out` is the compiled executable):

```
$./a.out FileA FileB
c: pippo.txt(1) pluto.txt(1)
bbbb: pippo.txt(2)
d1234: pluto.txt(1)
$
```

N.B.: The format of the output is just an example