



32

Mutant mushrooms

30 points

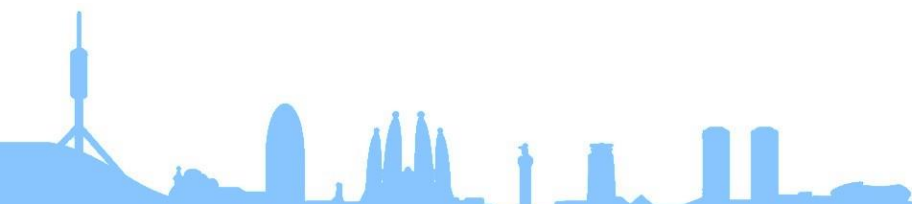
Introduction

Doctor Crazyus Maximus has found a mechanism to manipulate the DNA of mushrooms to make them replicate at a very fast rate, but this genetic manipulation shows a very strange effect: depending on the mutation type, the mushrooms are replicated following specific patterns.

After some experimentation, Crazyus found 4 relevant facts:

1. When mushrooms replicate, they do not fill a space which is already filled with some other mushroom of the same species.
2. The mutant mushrooms replication is quite aggressive, and they destroy other species. It seems that there is one gene of the mutation process that is determining the resilience of the mushroom, thus the mushrooms with higher resilience are the ones that prevail in case of direct contact. You can assume that different mushrooms will never have the same resilience.
3. The growth of the mutant mushrooms is very fast, but also their death. It seems that there is another gene that defines how many days a mushroom will be alive, with no error. Luckily, the "age" of the mushroom is not inherited when the mushroom is replicated, so new mushrooms can have their own life, starting from 0.
4. Once a mushroom passes its maximum age, it stops replicating and dies, leaving the space empty at the end of the day. But the mushroom will still be present during the day, so surrounding mushrooms of the same type won't be able to replicate on it.

Can you help Doctor Crazyus to study the mutant mushrooms with your program to simulate their behaviour?





Input

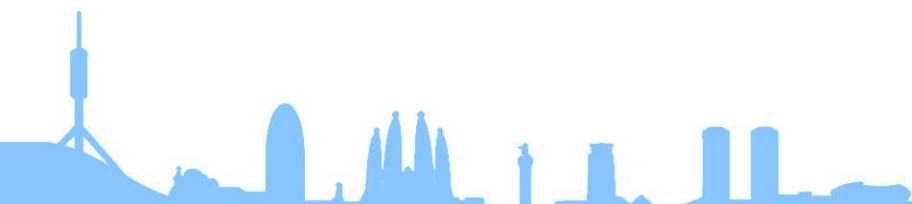
The input is structured as follows:

- A line with a positive integer, indicating how many species of mutant mushrooms will be simulated.
- For each of the mushroom species:
 - o A line with a character representing the label for the mushroom.
 - o A line with 2 positive integers, the first one being the resilience of the mushroom and the second one the number of days of life for the mushroom. To avoid having a dirty output, the number of days is limited in the range [0, 9].
 - o A line with 2 positive integers, being the first one the rows of the mushroom growth pattern and the second one the columns of the pattern.
 - o The growth pattern, represented by a grid of 0s and 1s.
- A line with 2 positive integers: the rows and the columns of the simulated experiment.
- The simulated experiment, which is formed by a regular grid according to the provided dimensions. The map indicates the initial places for the mushroom species (it may be more than one initial place for each species). The character '.' indicates an empty space that can be covered by a mushroom, and the character '_' indicates a space outside the simulation (so it must never be covered).
- A line with a positive integer indicating the number of days for the simulation.

NOTE: When applying the mushroom growth pattern, the mushroom is always in the center. So, you can assume that the growth pattern dimensions are always odd numbers.

Output

The program must show the status of the simulation on each day, representing all the mushrooms that are alive and their age (see the example below).

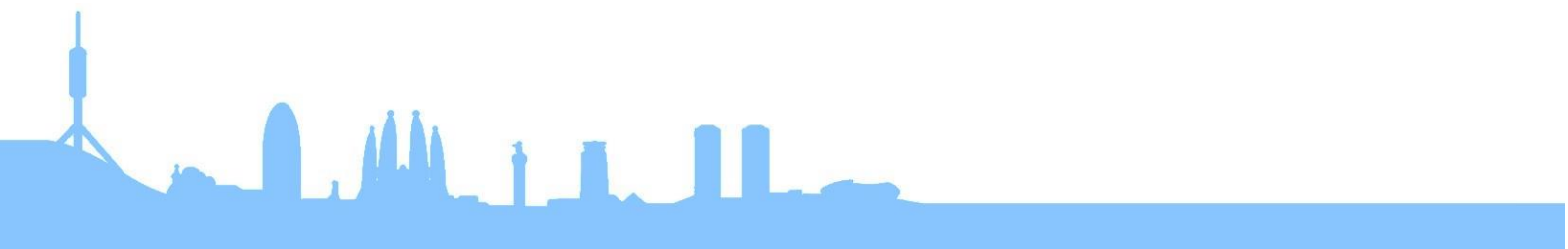




Example

Input

```
3
A
30 4
3 3
010
101
010
C
20 5
5 5
10001
01010
00000
01010
10001
L
10 6
3 5
11000
10000
10011
12 22
A....._.....L
....._.....
....._....._.....
....._.....
....._.....
....._.....
....._.....
....._.....
....._.....
....._.....
```





....._.....

....._.....

C....._.....L

6

Output

Mushrooms at day 0

A....._.....L 0....._.....0

....._.....

....._.....

....._.....

....._.....

....._.....

....._.....

....._.....

....._.....

....._.....

....._.....

C....._.....L 0....._.....0

Mushrooms at day 1

AA....._...L.L 10....._...0.1

A....._...L.. 0....._...0..

....._.....

....._.....

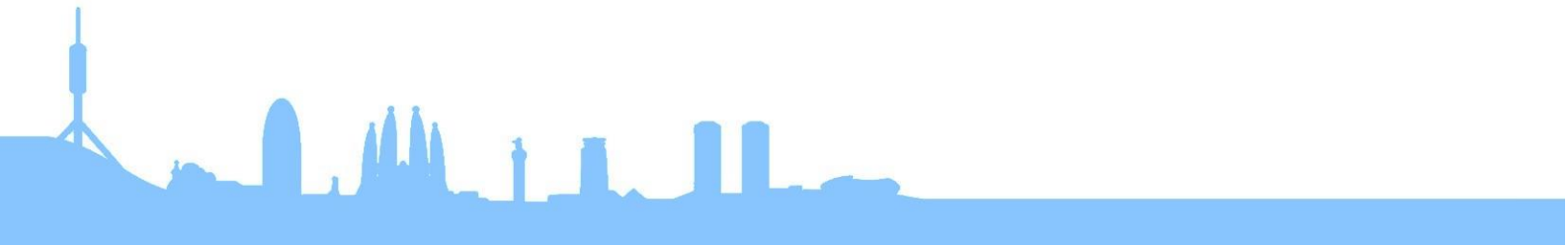
....._.....

....._.....

....._.....

....._.....

....._.....





```

..C.._..... ..0.._.....
.C....._.....LL. .0....._.....00.
C....._.....L.L 1....._.....0.1

```

Mushrooms at day 2

```

AAA....._..LLL.L 210....._..001.2
AA....._..L.LLL 10....._..0.100
A....._....._..L..LL 0....._....._..0..00
....._.....
....._.....
....._.....
....._.....
C...C....._..... 0...0....._.....
.C.C._..... .0.0._.....
C.C._.....LLL.. 0.1._.....000..
.C.C....._.....LLLL. .1.0....._.....0011.
C.C.C._.....LLLLL 2.0.0._.....00102

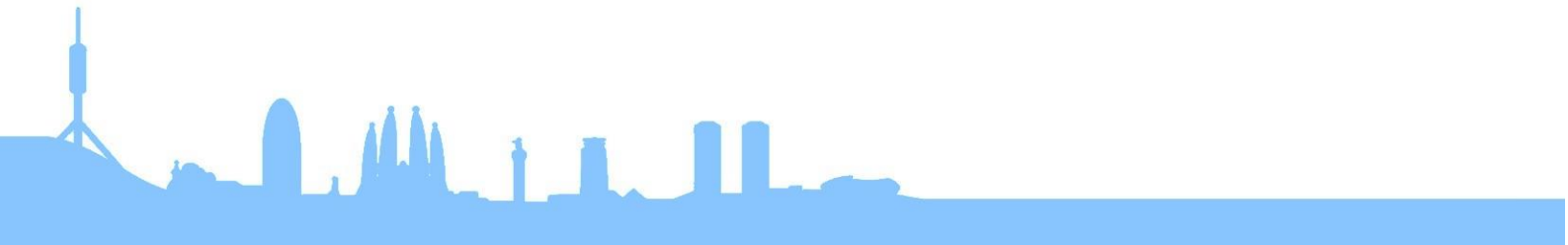
```

Mushrooms at day 3

```

AAAA....._LLLLLL 3210....._011203
AAA....._LLLLLL 210....._010211
AA....._....._..LLLLL 10....._....._..10011
A....._.....L..LL.L 0....._....._..0..00.0
....._.....
..C...C....._..... ..0...0....._.....
.C.C.C....._..... .0.0.0....._.....
C.C.C....._..... 1.0.1....._.....
.C.C._.....LLLL... .1.1._.....0000...
C.C.C_.....LLLLL.. 1.2.0_.....00111..

```





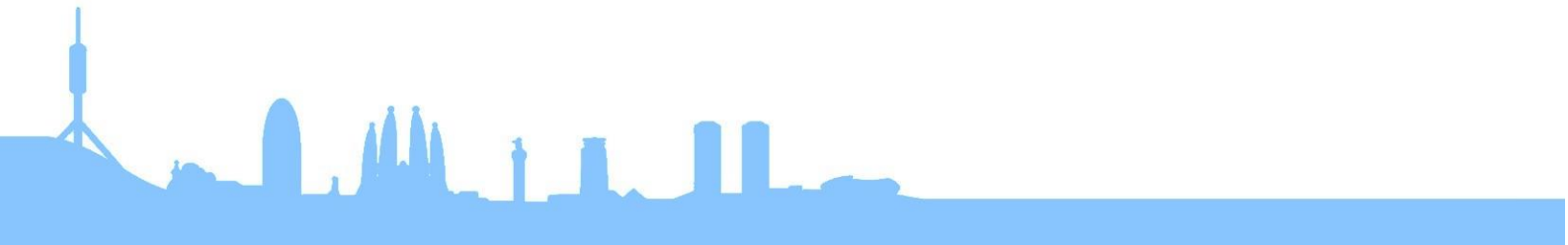
```
.C.C.C.....__..LLLLLLL .2.1.0.....__..0011220
C.C.C.____.....LLLLLLL 3.1.1.____.....0011213
```

Mushrooms at day 4

```
.AAAA.....L_LLLLLL .3210.....0_122314
AAAA....._LLLLLLL 3210....._121322
AAA...__.....L_LLLLLL 210...__.....0_021122
AA..C.__C....L.LLLLLLL 10..0.__0....0.1001101
AC.C.C.C.....L.._..LLL 00.0.0.0.....0.._.000
C.C.C.C.....__..... 0.1.0.1.....__.....
.C.C.C.C.____..... .1.1.1.0.____.....
C.C.C.C.C..._LLLL.... 2.1.2.0.0..._0000....
.C.C.__C.....LLLLLL... .2.2.__0.....001111...
C.C.C____.....LLLLLLLLL. 2.3.1____.....00112220.
.C.C.C....._LLLLLLLLLL .3.2.1....._001122331
C.C.C.C____.....LLLLLLLLLL 4.2.2.0____.....001122324
```

Mushrooms at day 5

```
..AAAA.....L.L_LLLLLL ..3210.....0.1_233425
.AAAA.C...CLL._LLLLLL .3210.0...000._232433
AAAA.C__C.LLL_LLLLLL 3210.0__0.001_132233
AAA.C.__C..LLLLLLLLLLL 210.1.__1..00102112212
AA.C.C.C.C.L.LLL_LLLL 10.1.1.1.0.0.100_0111
A.C.C.C.C.CL..LL_LLLL 0.2.1.2.0.00..00_0000
.C.C.C.C.C_LLLLL..... .2.2.2.1.0_00000.....
C.C.C.C.C..L_LLLL.... 3.2.3.1.1..0_1111....
.C.C.__C.C.LLLLLLLLLL.. .3.3.__1.0.001122220..
C.C.C__C.C.LLLLLLLLLLL. 3.4.2__0.00011223331.
.C.C.C.C.C._LLLLLLLLLL .4.3.2.0.0._112233442
```





..C.C.C__..LLLLLLLLLLLLL ..3.3.1__..00112233435

Mushrooms at day 6

...AAAAC.CLCL.L_LLLLLL. ...32100.0001.2_34453.
 ..AAAAC.CLCLLL_LLLLLL ..32101.001110__343544
 .AAAAC__.CLCLL_LLLLLL .32101__.10012__243344
 AAAAC.___CLCLLLLLLLLLLL 32102.___20010213223323
 AAAC.C.C.CLCLLLL_LLLL 2102.2.2.1000211__1222
 AAC.C.C.CLCLLLL_LLLL 103.2.3.10110011__1111
 AC.C.C.C.C_LLLLLLLLLL 03.3.3.2.1__1111100000
 C.C.C.C.CLCL_LLLLLL... 4.3.4.2.2001__22220...
 .C.C.___C.CLCLLLLLLLLLL.. .4.4.___2.10012233331..
 C...C__.CLCLLLLLLLLLLL 4...3__.10111223344420
 ...C.C.C.CL_LLLLLLLLLL ...4.3.1.10__223344553
 ..C.C.C_LCLCLLLLLLLLLL. ..4.4.2__001022334454.

