CS106X
Autumn 2019

Handout 1139
October 7^{th}, 2019

Star Wars, Blenjeel Worms, and Color Wriggles

The Blenjeel sand worms slither beneath the sandy surface of the planet Blenjeel. As the planet's only known inhabits, they protect their homeland by attacking and devouring anyone who steps foot on their planet.

Of course, sand worms need to be strong, flexible, and able to slither as quietly as possible. All teenage sand worms are conscripted to a six-month boot camp, where they endure intense training. The most demanding of the training exercises is the famous "wriggle test", which requires worm cadets to slither from one position to a parallel position several hundred yards away. Only the toughest, most determined of worms survive.

The exercise is so famous that the Jedi Academy's CS106X courses have its students solve a puzzle based on it. That puzzle goes something like this:

Color Wriggles

Given an n by m board (3 <= n <= 6, 5 <= m <= 50) wriggle a Blenjeel sand worm (of length n) from the left column to the right column, ensuring the worm never occupies two squares of the same color.

Consider the following 3 by 5 board, where each number represents a different color, and the worm is represented by the chain of circles:

1	2	3	2	4
3	1	3	1	2
4	1	4	3	1

One of the two possible moves? Pull the bottom of the worm to the right so it's positioned as follows:

1	2	3	2	4
3	1	3	1	2
4	1	4	3	1

Now we can pull from the **other** end of the worm to carry it through three different positions:

1	2	3	2	4
3	1	3	1	2
4	1	4	3	1

1	2	3	2	4
3	1	3	1	2
4	1	4	3	1

1	2	3	2	4
3	1	3	1	2
4	1	4	3	1

Through a series of additional moves, it's possible to wriggle the worm into the target position:

1	2	3	2	4
3	1	3	1	2
4	1	4	3	1

It's acceptable for the worm to reach the right column in either orientation.

Your job is to write a program that reads in a series of boards as described above, and for each prints out the **minimum** number of wriggles needed to move the worm from the left column to the right (or -1 if there's no path between the two).

Input

The data needs to be read in from a file named **`color-wriggles.in`**, and the format of that file can be gleaned from the following sample file:

```
12324
31312
41431

234234
342112
421311

234233
342112
421331

413444111221344412 31
223134334143233123 12
122312213121241433 23

41251355234115
13515533543252
34212412323543
52454355242421

364311121136362
151446122112155
434232633624623
561614315456464
234426516251346

end
```

Each board is separated from the next by a blank line, and a standalone **`end`** will mark the end of input. Again, the filename **`color-wriggles.in`** should be hard coded into your program, and your program otherwise shouldn't interact with standard input at all.

Output

For each board read from **`color-wriggles.in`**, print (to **`cout`**, with no gratuitous whitespace) the minimal number of wriggles needed to move the worm from the left column to right.

```
16
17
-1
39
31
58
```

Incentive

Everyone who submits efficient, working solutions will be treated to dinner with Jerry and Garrick at a time that's convenient for all. By efficient, I mean that all of the individual boards can be collectively solved in a matter of seconds. By working, I mean that you read data in from a file named **`color-wriggles.in`** (I don't supply one, so you'll create one, and I'll use my own when testing yours) and prints the answers out—one per line—to standard output using **`cout`**.

You should email, as an attachment, a single file that compiles in Qt Creator and solves this problem. This C++ file should be a full working program that depends only on standard and CS106X libraries.

If you submit an incorrect solution, I'll let you know and let you resubmit again and again until you either succeed or give up.

.