

---

Article

[Danny Wijnschenk](#) · Nov 8, 2017 3m read

## Advent of Code 2016 Day8: Two-Factor Authentication

This is a series of programming challenges for beginners and experienced Caché programmers.

For an introduction : go to article <https://community.intersystems.com/post/advent-code-2016-day1-no-time-ta...>

The challenge of today has nothing to do with real two-factor authentication ! (sorry if you came to this article by searching the real thing)

For the complete explanation of the challenge, go to <http://adventofcode.com/2016/day/8>.

Your input (<http://adventofcode.com/2016/day/8/input>) consists of a series of three basic instructions to control the display of a screen.

The goal of the first part of the challenge is to calculate how much pixels will be lit after all instructions are executed.

The three basic instructions are :

```
rect AxB : turns all pixels on, as a top-left rectangle A wide and B tall
rotate row y=A by B : shifts all the pixels in row A right by B pixels
rotate column x=A by B : shifts all the pixels in column A down by B pixels
```

(screen coordinates are zero based, pixels that would fall off end up at the other end)

The second part of the puzzle is to find out what is written on the screen.

To make it a bit more interesting, the solution displays the screen while it is being build by each instruction.

For this challenge, Bert and I almost wrote the same code, using an array to hold the screen pixels to do the 3 basic instructions. Here is the code :

```
Class AOC2016.Day8 Extends AOC2016.Utils
{

ClassMethod Part1(file As %String = "day8.txt")
{
    #Dim input, instruction, command as %String
    #Dim param1, param2 as %Integer
    #Dim iInput, x, y as %Integer
    #Dim screen, screenCopy as %String
    #Dim maxX as %Integer = 50
    #Dim maxY as %Integer = 6
    #dim count as %Integer = 0

    Try {
        //Initialize screen display
```

```

For x = 0:1:maxX-1 {
  For y = 0:1:maxY-1 {
    Set screen(x,y) = " "
  }
}
Do ..Input(file, .input)
For iInput=1:1:input {
  Set instruction = $ZStrip(input(iInput),"<>W")
  Set command = $Piece(instruction," ",1)
  If command="rect" {
    //Draw square in upper left corner
    Set param1=$Piece($Piece(instruction," ",2),"x",1)
    Set param2=$Piece($Piece(instruction," ",2),"x",2)
    For x=0:1:param1-1 {
      For y=0:1:param2-1 {
        Set screen(x,y)="#"
      }
    }
  } else {
    Set command = $Piece($Piece(instruction," ",1,3),"=",1)
    Set param1 = $Piece($Piece(instruction," ",3),"=",2)
    Set param2 = $Piece(instruction," ",5)
    //rotate pixels down
    If command = "rotate column x" {
      Merge screenCopy=screen(param1)
      For y=0:1:maxY-1 {
        Set screen(param1,(y+param2)#maxY) = screenCopy(y)
      }
    }
    //rotate pixels right
  } elseif command = "rotate row y" {
    Set screenCopy="" For x=0:1:maxX-1 { set screenCopy=screenCopy_screen(x,
param1) }
    For x=0:1:maxX-1 {
      Set screen((x+param2)#maxX,param1) = $E(screenCopy,x+1)
    }
  }
}
//Display screen for every step
Hang .1 //for some animation effect
Write # //clear the screen
For y=0:1:maxY-1 {
  For x=0:1:maxX-1 {
    Write screen(x,y)
  }
  Write !
}
}
//Count leds in display
For y=0:1:maxY-1 {
  For x=0:1:maxX-1 {
    If screen(x,y)="#" Set count=count+1
  }
}
} Catch {
  Write "Error : ",$ZE,!
}
Write "Count : ",count,!
}

```

```
}
```

Look here for all our solutions so far : <https://bitbucket.org/bertsarens/advent2016> and <https://github.com/DannyWijnschenk/AdventOfCode2016>

Here is the list of all Advent of Code 2016 articles :

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#)

[#Caché](#) [#Code Snippet](#) [#Contest](#) [#ObjectScript](#)

---

Source URL: <https://community.intersystems.com/post/advent-code-2016-day8-two-factor-authentication>