
Article

[Danny Wijnschenk](#) · Nov 2, 2017 4m read

Advent of Code 2016 Day2: Bathroom Security

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

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

In this challenge, you need to find a password using instructions to move on a keypad.
Instructions can be U(p), D(own), L(ef) and R(ight).

You start at button 5 on a keypad like

```
1 2 3
4 5 6
7 8 9
```

each line of instructions lead to one digit of the password.

Full description can be found at <http://adventofcode.com/2016/day/2>

For example : ULL leads to '1' (a move outside the keypad can be ignored)

Using the full input file found at <http://adventofcode.com/2016/day/2/input>, you have to compose the complete password.

I find this puzzle easier than the previous one : for each line in input, looping through each instructions, we can easily calculate for each heading, for each button what the next button should be.

Lets use for each heading a \$Case to find for a button what the next button will be :

For example, for heading U(p), when on button 1, 2 or 3, the 'up' one is the same (out of keypad). When on button 4, the 'up' one is 1, when on button 5 the 'up' one is 2, and so on :

If heading="U" Set button = \$Case(button, 1:1, 2:2, 3:3, 4:1, 5:2, 6:3, 7:4, 8:5, 9:6)

Do this for each heading, for each instruction and done !

Again, try to make this first before looking at a possible solution.

Here is one solution :

```
ClassMethod Part1(file As %String = "j:\winfo\aac\day2.txt")
{
#Dim keypad as %String = ""      ;all the buttons after each line of instructions
#Dim button as %Char = "5"      ;current button
#Dim heading as %Char           ;instruction heading
```

```

#Dim line as %String
#Dim iPiece as %Integer

Open file:"R":1 Else Use 0 Write "Could not open file ",file Quit
Try {
  For {
    Use file Read line If line="" Quit
    Set line=$TR(line," ")
    For iPiece=1:1:$Length(line) {
      Set heading = $E(line,iPiece)
      If (heading="U") {
        Set button=$Case(button,1:1, 2:2, 3:3, 4:1, 5:2, 6:3, 7:4, 8:5, 9:6)
      } elseif (heading="D") {
        Set button=$Case(button,1:4, 2:5, 3:6, 4:7, 5:8, 6:9, 7:7, 8:8, 9:9)
      } elseif (heading = "R") {
        Set button=$Case(button,1:2, 2:3, 3:3, 4:5, 5:6, 6:6, 7:8, 8:9, 9:9)
      } elseif (heading = "L") {
        Set button=$Case(button,1:1, 2:1, 3:2, 4:4, 5:4, 6:5, 7:7, 8:7, 9:8)
      }
    }
    Set keypad = keypad _ button
  }
} Catch {
  If $ZE'["ENDOFFILE" Use 0 Write "Error : ",$ZE,!
}
Close file
Use 0 Write "Password : ",keypad,!

Quit
}

```

If you run this code, and enter the password, this will unlock the second part of today's challenge.

The second part of the challenge gives us the same puzzle, but the keypad is different :

```

1
2 3 4
5 6 7 8 9
A B C
D

```

Turns out that we need to only change our four \$Case statements to reflect the different layout.

We could create a new challenge where we would allow any keypad layout as input, turns out that Bert (the co-author of this series) did just that.

Here is his code :

```

Start() PUBLIC {
s input=""
/*
s keypad(1)="00000"
s keypad(2)="01230"
s keypad(3)="04560"
s keypad(4)="07890"
s keypad(5)="00000"

```

```

s xPos=3
s yPos=3
*/

s keypad(1)="0000000"
s keypad(2)="0001000"
s keypad(3)="0023400"
s keypad(4)="0567890"
s keypad(5)="00ABC00"
s keypad(6)="000D000"
s keypad(7)="0000000"

s xPos=2
s yPos=4

#Dim objFileStream As %Stream.FileCharacter
s objFileStream = ##Class(%Stream.FileCharacter).%New()
s sc=objFileStream.LinkToFile(
"C:\Users\15274\workspace\adventofcode\2016\input\day02\input.txt")
while 'objFileStream.AtEnd {
    s line=objFileStream.ReadLine()
    f i=1:1:$LENGTH(line) {
        s command=$EXTRACT(line,i)
        if ((command="R") && ($$getValueFromKeypad(.keypad,xPos+1,yPos)'=0)) {
            s xPos=xPos+1
        } elseif ((command="L") && ($$getValueFromKeypad(.keypad,xPos-1,yPos)'=0)) {
            s xPos=xPos-1
        } elseif ((command="U") && ($$getValueFromKeypad(.keypad,xPos,yPos-1)'=0)) {
            s yPos=yPos-1
        } elseif ((command="D") && ($$getValueFromKeypad(.keypad,xPos,yPos+1)'=0)) {
            s yPos=yPos+1
        }
    }
    w !,$$getValueFromKeypad(.keypad,xPos,yPos)
}
//w !,$$getValueFromKeypad(.keypad,xPos,yPos)
}

getValueFromKeypad(KeyPad,XPos,YPos) {
    q $EXTRACT(KeyPad(YPos),XPos)
}

```

And another two stars and a nicer advent calendar on <http://adventofcode.com/2016>.

Look here for our solutions : <https://github.com/DannyWijnschenk/AdventOfCode2016> : 2016Day2.xml and <https://bitbucket.org/bertsarens/advent2016>.

See you tomorrow !

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-day2-bathroom-security>