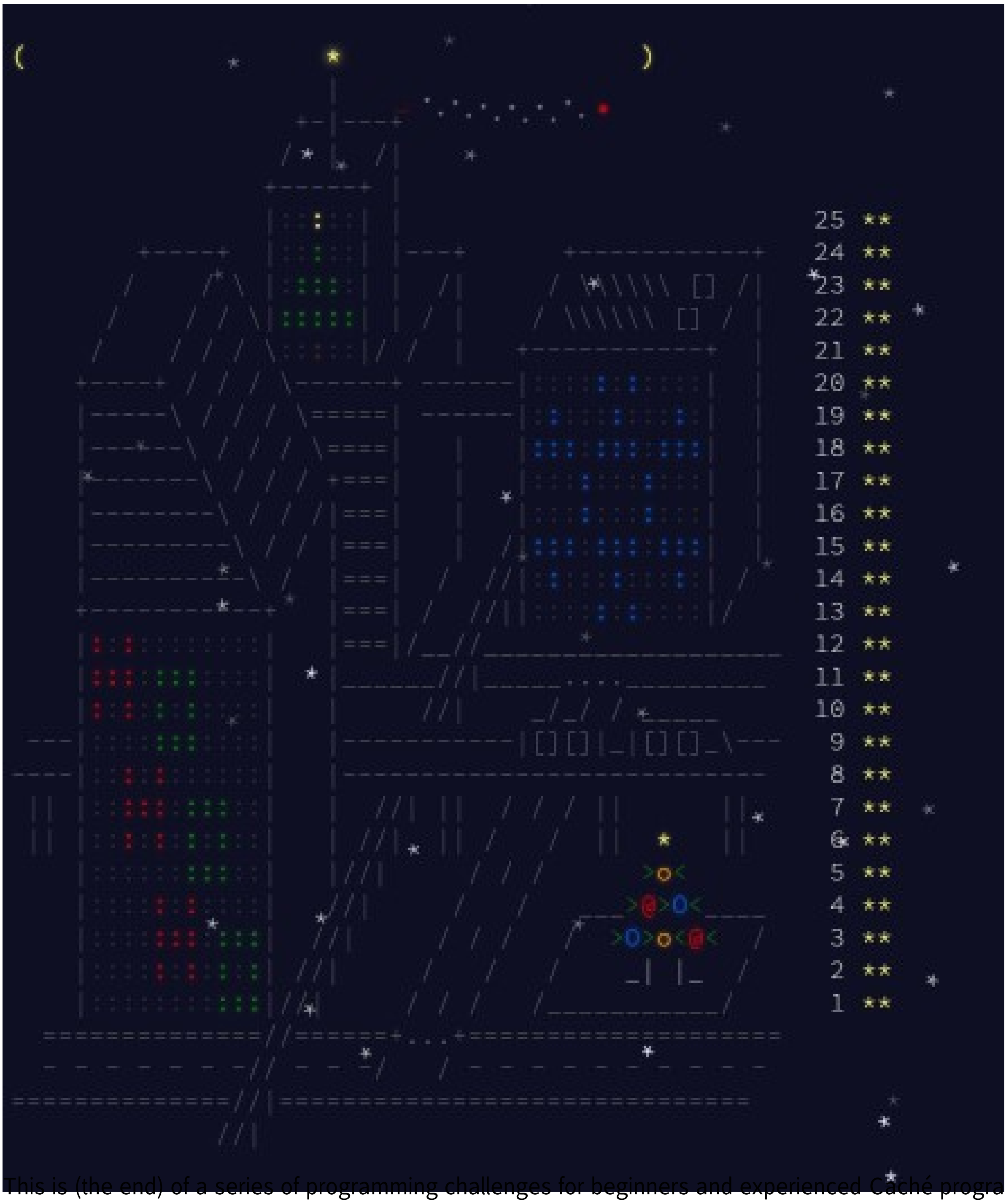
Article

Danny Wijnschenk · Nov 25, 2017 4m read

# Final Advent of Code 2016 Day25: Clock Signal



This is (the end) of 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...

This is the 25th and last challenge of the 2016 series of Advent of Code.

In this challenge, you have to send a signal out, using the assembunny code interpreter that we had to make in earlier challenges.

There is one extra instruction added to the assembunny language, which outputs a value (out x).

Using the instructions as input (see http://adventofcode.com/2016/day/25 for the link to your input file), what is the lowest positive integer in register **a** to cause the instructions to output an endless clock signal of 0, 1, 0, 1, 0... ?

Using much of the code we had to write for the assembunny in previous challenges (challenge 12 and challenge 23 ), i added the instruction out , and a test to make sure the outputted signal is a repeating pattern of 0 and 1.

Here is the code :

```
Class AOC2016.Day25 Extends AOC2016.Utils
{

ClassMethod Part(file As %String = "day25.txt", part As %Integer = 1)
{
  #Dim iInput as %Integer
  #Dim input, iTry as %String

  Try {
    Do ..Input(file, .input)
    For iTry=0:1 {
      Write iTry," : "
      If ..Try(.input,iTry) Quit
      Write !
    }
  } catch {
    Write "Error : ",$ZError,!
  }
  Write !,"lowest integer = ",iTry,!
}

ClassMethod Try(input, a = 0, b = 0, c = 0, d = 0) As %Boolean
{
  #Dim error as %Integer = 0
  #Dim out, lastOut as %Char = ""
  #Dim iInput as %Integer = 1
  #Dim count as %Integer = 0
  #Dim line, instruction as %String
  #Dim param1, param2 as %Integer
  #Dim register as Array of %Integer

  Set register("a")=a
  Set register("b")=b
  Set register("c")=c
  Set register("d")=d
  While iInput '> input {
    Set line = input(iInput)
    Set instruction=$Piece(line," ",1)
    Set param1=$Piece(line," ",2)
    Set param2=$Piece(line," ",3)
    If instruction = "cpy" {
      Set register(param2)=$Select($Data(register(param1)):register(param1),1:param1)
    } elseIf instruction = "inc" {
      Set register(param1) = register(param1) + 1
    } elseIf instruction = "dec" {
      Set register(param1) = register(param1) - 1
    } elseIf instruction = "jnz" {
      If $Select($Data(register(param1)):register(param1),1:param1)'=0 {
        Set iInput = iInput + param2 -1 //-1 since every loop will do +1 anyway
      }
    } elseIf instruction = "out" {
      set out = $Select($Data(register(param1)):register(param1),1:param1)
      write out
      If (lastOut '= ""), ((lastOut=1)&(out'=0)) ! ((lastOut=0)&(out'=1)) Set
error = 1
      set lastOut = out
      Set count = count + 1
    }
    Set iInput = iInput + 1
    If count > 500 Quit
//forever takes too long to compute, 500 is enough eternity ;)
    If error Quit
  }
  Quit 'error
}

}
```

After submitting the correct answer, you will get two possible outcomes :

- if you did not submit a correct answer in all previous challenges, you are invited to do so, and you are stuck here.

- if you have submitted correctly to all the 2016 challenges, the quest is finished, the adventures ceased to exist, the journey stopped, the undertaking concluded,  the ordeal is no more !!!

The screen fills itself with congratulations, stars and snowflakes, and the advent calendar is also animated with stars and flying reindeer.

But most of all, it is a relief that the series of challenges successfully ended ... for 2016.

If you think you are trained enough, just wait a few days and go to the  Advent of Code website, because the 2017 challenge will start on December 1st.

Thanks for watching all the challenges, and don't forget to submit your code, we can all learn from it !

Look here for all our solutions we submitted  : https://bitbucket.org/bertsarens/advent2016 and https://github.com/DannyWijnschenk/AdventOfCode2016

List of other Advent of Code 2016 challenges so far :
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/final-advent-code-2016-day25-clock-signal