
Article

[Danny Wijnschenk](#) · Nov 12, 2017 3m read

Advent of Code 2016 Day12: Leonardo's Monorail

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...>

Today, you need to make a compiler for the language assembunny.

Luckily, this language has only 4 instructions :

cpy x y copies x (either an integer or the value of a register) into register y.

inc x increases the value of register x by one.

dec x decreases the value of register x by one.

jnz x y

jumps to an instruction y away (positive means forward; negative means backward), but only if x is not zero.

For example :

```
cpy 41 a
inc a
inc a
dec a
jnz a 2
dec a
```

After executing these instructions, the register **a** would contain 42.

Full explanations are on <http://adventofcode.com/2016/day/12>, you will find your input at <http://adventofcode.com/2016/day/12/input>.

After executing all instructions of your input, what will be the contents of register a ?

No big challenges in this exercise : we can start at the first instruction, execute each instruction (set a register or jump a few instructions) and move 1 instruction forward (except for instruction jnz).

When we executed the last one, we can display the contents of register a.

Here is the code (Bert and I had similar code) :

```
Class AOC2016.Day12 Extends AOC2016.Utils
{
```

```
ClassMethod Part(file As %String = "day12.txt", part As %Integer = 1)
{
```

```
#Dim iInput as %Integer
#Dim input,instruction, param1, param2 as %String
#Dim var, register, line as %String

For var="a","b","c","d" Set register(var)=0
If part'=1 set register("c")=1 ;only for challenge 2

Try {
  Do ..Input(file, .input)
  Set iInput = 1
  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
      }
    }
    Set iInput = iInput + 1
  }
} catch {
  Write "Error : ", $ZError, !
}
Write "register a = ", register("a"), !
}

}
```

After submitting our answer, the second challenge is unlocked : we just need to initialize register c to 1, which will cause the instructions to loop a lot more, but the execution time is not bad.

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-day12-leonardos-monorail>
