
Article

[Danny Wijnschenk](#) · Nov 9, 2017 4m read

Advent of Code 2016 Day9: Explosives in Cyberspace

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's challenge is about decompressing input that is compressed in an experimental format.
In the format, markers indicate how much time a number of characters need to be repeated.

For example :

A(1x5)BC
repeats only the B a total of 5 times, becoming ABBBBBC for a decompressed length of **7**.
(3x3)XYZ becomes XYZXYZXYZ for a decompressed length of **9**.
A(2x2)BCD(2x2)EFG doubles the BC and EF, becoming ABCBCDEFEFG for a decompressed length of **11**.
(6x1)(1x3)A simply becomes (1x3)A
- the (1x3) looks like a marker, but because it's within a data section of another marker, it is not treated any differently from the A that comes after it. It has a decompressed length of **6**.
X(8x2)(3x3)ABCY becomes X(3x3)ABC(3x3)ABCY
(for a decompressed length of **18**)
, because the decompressed data from the (8x2) marker (the (3x3)ABC) is skipped and not processed further.

For the complete explanation of the challenge, go to <http://adventofcode.com/2016/day/9>.

Your input is found here : <http://adventofcode.com/2016/day/9/input>, you have to output the total length of the decompressed input.

Here is Bert's code as a routine :

```
Start() PUBLIC {
    #Dim objFileStream As %Stream.FileCharacter
    s objFileStream = ##Class(%Stream.FileCharacter).%New()
    s sc=objFileStream.LinkToFile(
"C:\Users\15274\workspace\adventofcode\2016\input\day09\input.txt")

    while 'objFileStream.AtEnd {
        s line=objFileStream.ReadLine()
        s result =$$decompressCount(line)
        w !,"part 1:",*9,result
    }
}

decompressCount(Data) PUBLIC {
    s count=0
```

```

for i=1:1:$LENGTH(Data) {
  if $EXTRACT(Data,i)="(" {
    s x=$FIND(Data,")",i+1)
    s first=$EXTRACT(Data,i+1,x-2)
    s letters=$PIECE(first,"x")
    s times=$PIECE(first,"x",2)
    s next=$EXTRACT(Data,x,x+letters-1)
    s i=x+letters-1
    s count=count+(letters*times)
  } else {
    s count=count+1
  }
}
q count
}

```

The second part of the challenge is a next version of the compress format : if there are markers in the decompressed section, they have to be decompressed as well.

For example :

(3x3)XYZ still becomes XYZXYZXYZ, as the decompressed section contains no markers.
 X(8x2)(3x3)ABCY becomes XABCABCABCABCABCBCY
 , because the decompressed data from the (8x2) marker is then further decompressed, thus triggering the (3x3) marker twice for a total of six ABC sequences.
 (27x12)(20x12)(13x14)(7x10)(1x12)A
 decompresses into a string of A repeated **241920** times.
 (25x3)(3x3)ABC(2x3)XY(5x2)PQRSTX(18x9)(3x2)TWO(5x7)SEVEN becomes **445** characters long.

What is length of the decompressed input using this decompression version 2 ?

Here is my code in a class that can decompress the two versions : by using recursion, we can reuse much of the code of the first part :

```

Class AOC2016.Day9 Extends AOC2016.Utils
{
  ClassMethod Part(file As %String = "day9.txt", version As %Integer = 1)
  {
    #Dim input as %String
    #Dim iInput as %Integer
    #Dim totalLength as %Integer = 0
    Try {
      Do ..Input(file, .input)
      For iInput=1:1:input {
        Set totalLength=totalLength+..Decompress(input(iInput), version)
      }
    } Catch {
      Write "Error : ", $ZError, !
    }
    Write "Total length : ", totalLength, !
  }
}

```

```
ClassMethod Decompress(line, version As %Integer = 1) As %Integer
{
    #Dim totalLength as %Integer = 0
    #Dim marker as %String
    #Dim length, times, firstPos, lastPos as %Integer
    For {
        If line="" Quit
        Set firstPos=$Locate(line,"\\d*x\\d*\\",,lastPos,marker)
        If firstPos>0 {
            set totalLength=totalLength+firstPos-1
            set length=+$E(marker,2,*)
            set times=+$Piece(marker,"x",2)
            Set line=$Extract(line,lastPos,*)
            Set totalLength=totalLength+(times*$Select(version=1:length,1:..
Decompress($Extract(line,1,length),version)))
            Set line=$Extract(line,length+1,*)
        } else {
            set totalLength=totalLength+$length(line)
            set line = ""
        }
    }
    Quit totalLength
}
```

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-day9-explosives-cyberspace>