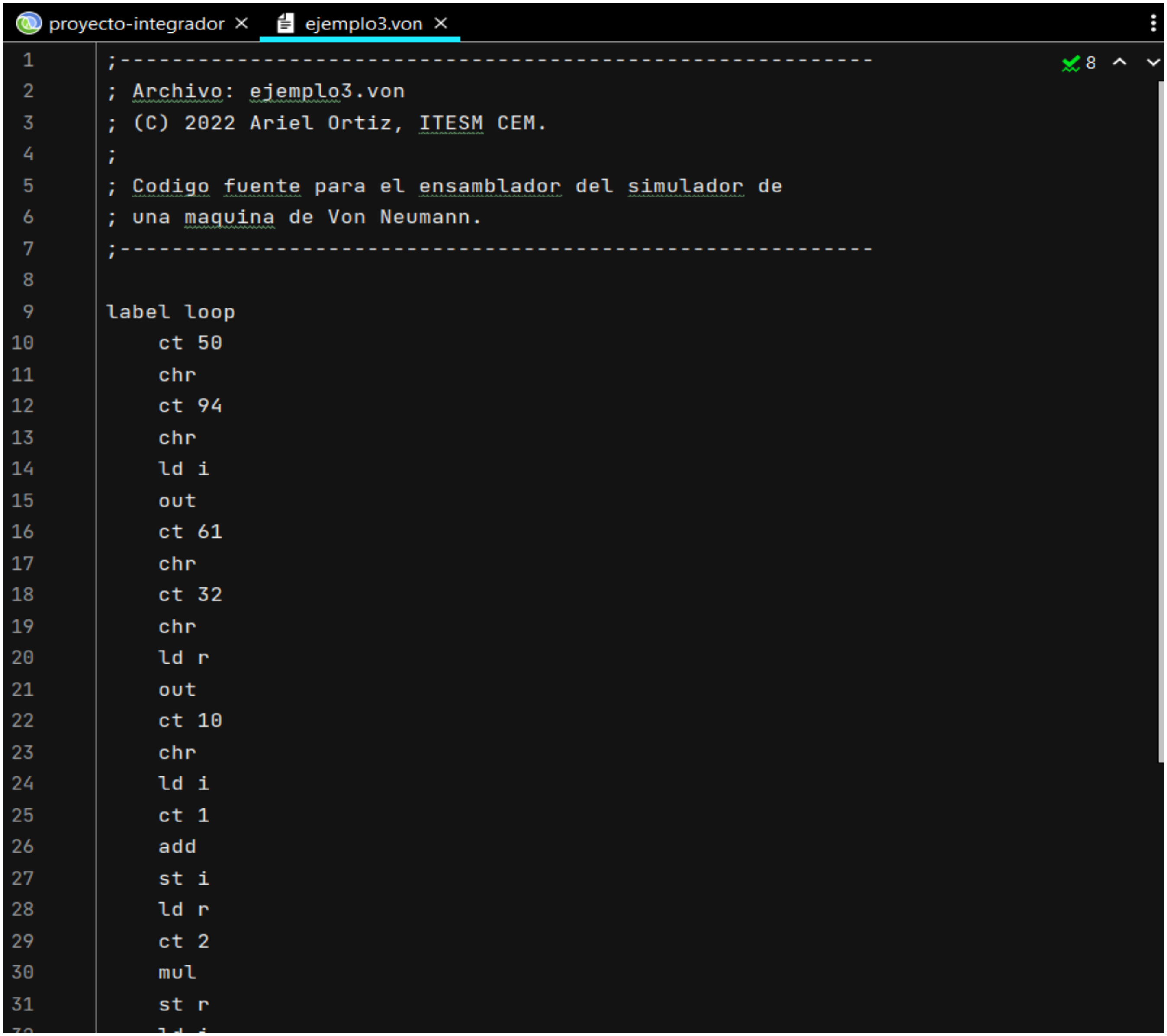
Clojure assembler simulation based on opcodes

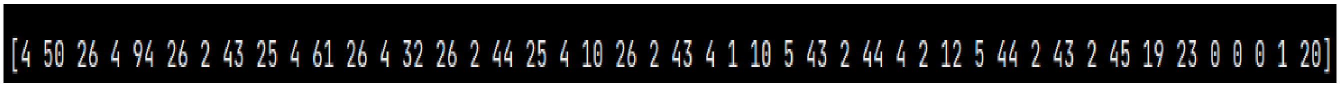
Authors: José Luis Madrigal Sánchez and Erika Marlene García Sánchez

This is one of the test files, which has a code that displays powers of number two till twenty



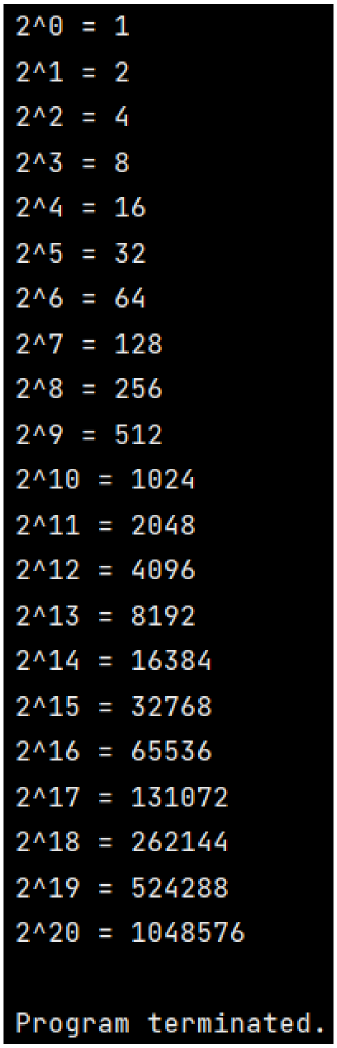
```
;----------------------------------------------------------
; Archivo: ejemplo3.von
; (C) 2022 Ariel Ortiz, ITESM CEM.
;
; Codigo fuente para el ensamblador del simulador de
; una maquina de Von Neumann.
;----------------------------------------------------------

label loop
    ct 50
    chr
    ct 94
    chr
    ld i
    out
    ct 61
    chr
    ct 32
    chr
    ld r
    out
    ct 10
    chr
    ld i
    ct 1
    add
    st i
    ld r
    ct 2
    mul
    st r
```

We pass the script to a vector, which makes the function of a stack, so then we can make the corresponding changes with labels and data. Then the commands (ct, add, ld…) are replaced with their opcodes (established by the teacher).



```
[4 50 26 4 94 26 2 43 25 4 61 26 4 32 26 2 44 25 4 10 26 2 43 4 1 10 5 43 2 44 4 2 12 5 44 2 43 2 45 19 23 0 0 0 1 20]
```

Then having the correct replacement, we apply the operations based on the opcodes, so we can print the result.



```
2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256
2^9 = 512
2^10 = 1024
2^11 = 2048
2^12 = 4096
2^13 = 8192
2^14 = 16384
2^15 = 32768
2^16 = 65536
2^17 = 131072
2^18 = 262144
2^19 = 524288
2^20 = 1048576

Program terminated.
```

I must mention that syntax errors are checked too, like the type, repetition or lack of a variable.