// This file is part of www.nand2tetris.org

// and the book "The Elements of Computing Systems"

// by Nisan and Schocken, MIT Press.

// File name: projects/04/Fill.asm

Runs an infinite loop that listens to the keyboard input.When a key is pressed (any key), the program blackens the

screen,i.e. writes "black" in every pixel;the screen should remain fully black as long as the key is pressed. When no key

is pressed, the program clears the screen, i.e. writes"white" in every pixel;the screen should remain fully clear as long

as no key is pressed.

// Set colors: black & white.

@32767

D = A

D = D + 1

D = D + A

@black

M = D

@0

D = A

@white

M = D

// Note that KBD is adjacent to the last SCREEN RAM.

@KBD

D = A

@last

M = D

// The current SCREEN RAM address.

@current

(INPUT)

// Reset `current`.

@SCREEN

D = A

@current

M = D

// Print a black or white screen based on input.

@KBD

D = M

@WHITESCREEN

D; JEQ

@BLACKSCREEN

0; JMP

(BLACKSCREEN)

// if (@current == @last), the full screen has been drawn.

@current

D = M

@last

D = M - D

@INPUT

D; JEQ

// Load color: black.

@black

D = M

// Set RAM address.

@current

A = M

// Set RAM content.

M = D

// Update @current.

A = A + 1

D = A

@current

M = D

@BLACKSCREEN

0; JMP

(WHITESCREEN)

@current

D = M

@last

D = M - D

@INPUT

D; JEQ

// Load color: white.

@white

D = M

@current

A = M

M = D

A = A + 1

D = A

@current

M = D

@WHITESCREEN

0; JMP

F2:

// 24576为键盘的地址 16384-24575 刚好8K 为屏幕的地址

@24575

D = A

// R0存储屏幕最大地址

@0

M = D

// R1存储屏幕当前地址

@SCREEN

D = A

@1

M = D

(LOOP)

@KBD

D = M

@FILL

D;JGT

@CLEAR

0;JMP

(FILL)

// 判断屏幕是否为满

@0

D = M

@1

D = D - M

@LOOP

D;JLT

@1

// 缓存当前地址

D = M

// 将当前地址推入A

A = M

// 将地址对应的屏幕位置变黑

// 如果使用 1，翻译成二进制是 0000000000000001

// -1 翻译成二进制是 1111111111111111

// 如果表示在屏幕上，1 有 15 位是空白的，1位是黑的

// -1 则屏幕上的 16 位全是黑色

M = -1

// 当前地址+1

@1

M = D + 1

@LOOP

0;JMP

(CLEAR)

// 判断屏幕是否为空

@SCREEN

D = A

@1

D = D - M

@LOOP

D;JGT

@1

// 缓存当前地址

D = M

// 将当前地址推入A

A = M

// 将地址对应的屏幕位置变白

M = 0

// 当前地址-1

@1

M = D - 1

@LOOP

0;JMP