# Preparation

# Story 1: Draw the starting snake

## Requirement

Create a two-segment snake (head + tail) near the middle of the screen

The snake should be a green colour

## Techniques

The simulation provides ‘addressable video memory’, that runs from (Word) location 256 (0x100 in hex) for the top-left corner, to 1023 (0x3ff in hex) for the bottom-right.

Exercise

What does the following code produce?

mov r0,#0

str r0,256

str r0,1023

[Paste in a partial screenshot showing only the output window]

mov r0,#1, means ‘move into register 0, the *immediate value 0’. This is known as ‘immediate address mode. In this case, the value 0 represents the colour black (no colour).*

*Good practices: When the program starts register 0 should default to the value 0, but it is not safe to assume this, so we set it to 0 explicitly. This is equivalent to the practice of initialising all variables in a higher level language.*

*We can specify other colours using the same RGB (Red Green Blue) format as used when creating a web page. This is best specified in hex, so for example, 0x008844 results in a suitable hue of green for the snake.*

*So, to draw a two segment snake somewhere near the middle of the screen we could write:*

*constants:*

*mov r0,#0x008844*

*drawSnake:*

*str r0,271*

*str r0,272*

***[Peter: assembler does not accept direct addressing to the screen memory, I think it should]***

***Notice that we have added two labels: ‘constants:’ and ‘drawSnake:’***