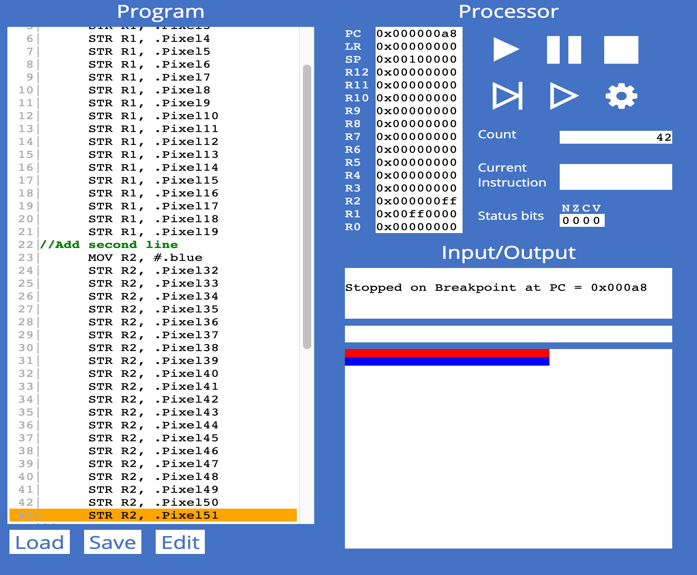
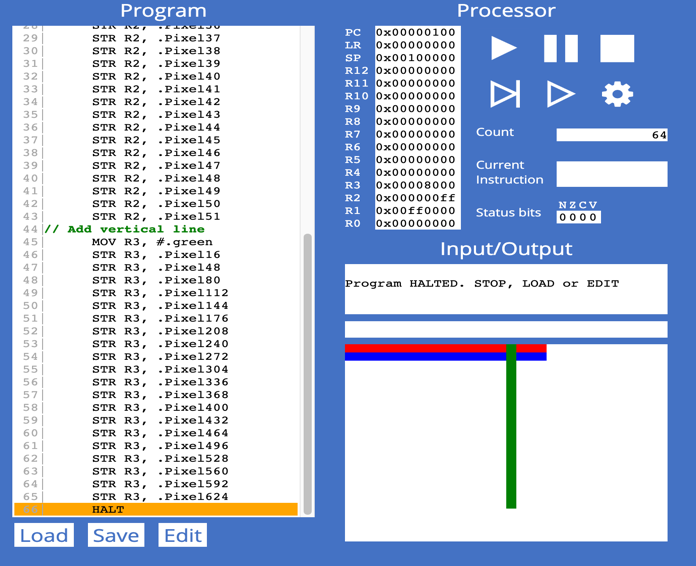
**COMPUTER SYSTEMS**

**LAB 09**

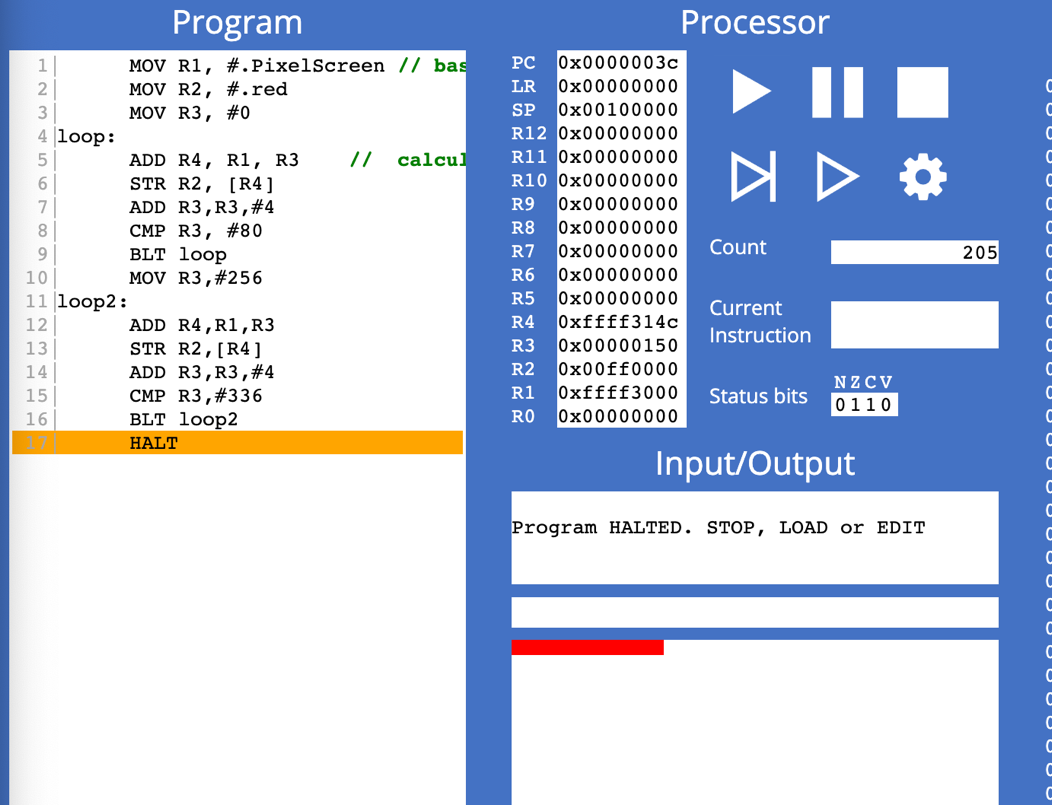






**Exercise 9.1.3**

1. The given code is indirect addressing as it can store the specific value of r2 in [r4] which has a memory address and in its own value. In the code it is adding four bytes every time to the pixel until it reaches 80 and moving data to the memory address 4 bytes, 32 bits. and R1 is the base address if we add value in R3 it will add to the R1 which stores the address and which can form the new pointer to the next pixel.



Graphical user interface, application

Description automatically generated

**Exercise 9.2.1**

Graphical user interface, application

Description automatically generated

**Exercise 9.2.2**

**Graphical user interface

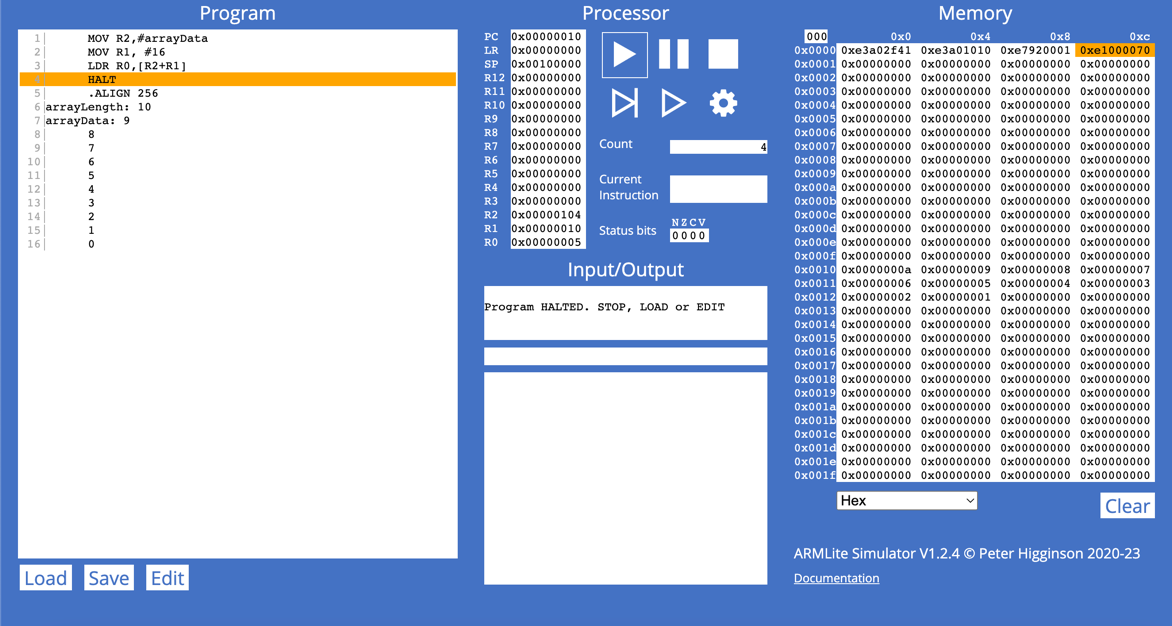
Description automatically generated**

**Exercise 9.3.1**

1. The .Align 256 instruction ensures the next instruction is aligned with a word address divisible by 256.

Graphical user interface, application

Description automatically generated

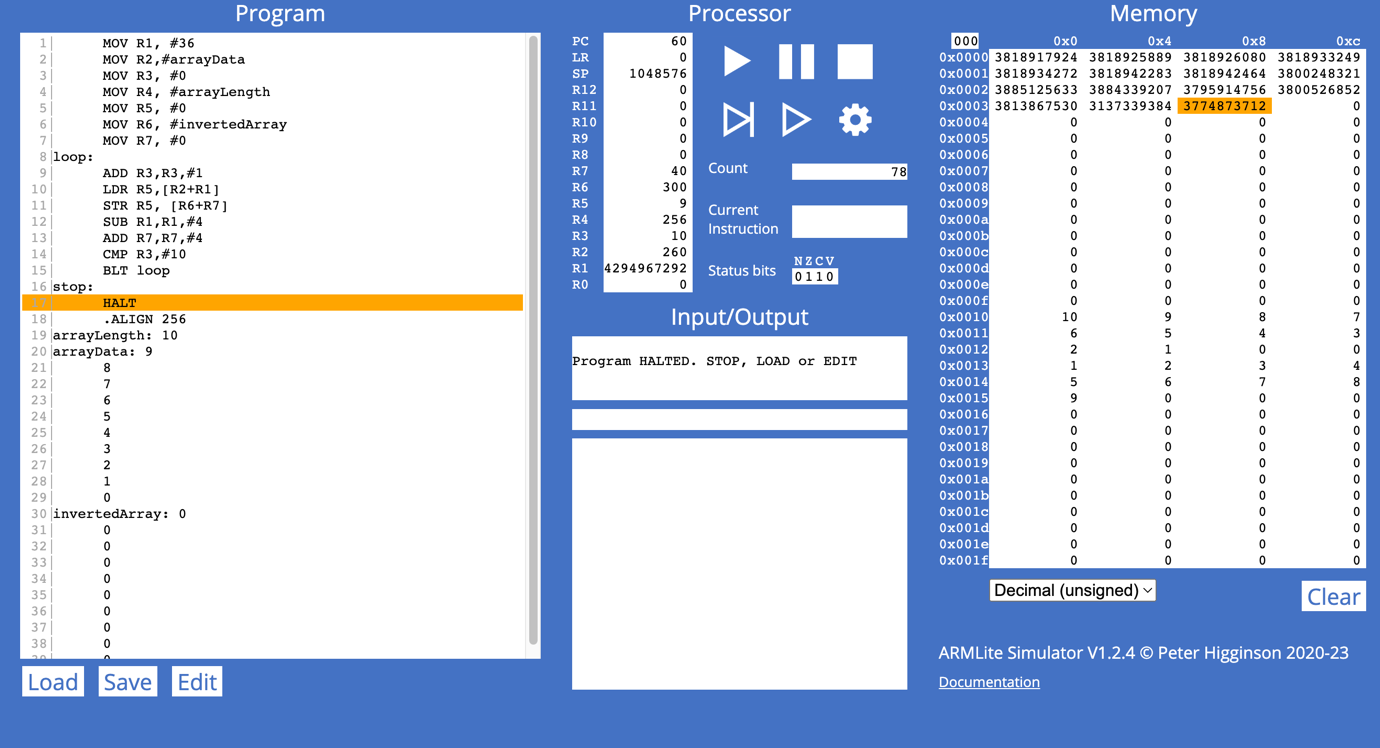


**Exercise 9.3.2**

Graphical user interface

Description automatically generated

**Exercise 9.4.1**

****

**Exercise 9.4.2**

**Graphical user interface, application

Description automatically generated**