1. Collaborators:

Riley Barklow:

* Designed and implemented the routine for dividing one binary number by another through repeated subtraction, leaving the modulo as the end output

Everett Holzapfel:

* Designed and implemented the routine for checking the binary numbers after each division to determine whether division needs to continue and when the final state is reached.
* Came up with the Turing machine itself: modulo of two binary numbers

1. Problem description:

The modulo operation is frequently used in programming and computer science, so

creating a Turing machine that outputs the modulo of two binary numbers seemed like an interesting way of seeing how a simple machine would compute modulo.

1. Annotated state diagram here:
2. Start specifications:

* The start state of the Turing machine is the first digit of the input provided by the user.
* The input for the turing machine should be two binary numbers of maximum value of 100, padded to 7 places with a ‘$’ separating them.
* The left binary number should be on the left of the modulo and the right binary on the right side.
* Example:
  + 7 % 3 = 0000111$0000011
  + 99 % 9 = 1100011$0001001
* What remains on the tape after execution is the output in binary