Isabelle Coletti
CSCI 360 Section 2
Spring 2022

1. One way to compute powers of 2 using only LA, MR, and the required even-odd register pair is to begin by loading 2 into the odd-register of the pair. If using the even-odd register pair of 2 and 3, this would look like LA 3,2. Then I would use MR to multiply the 2 by itself. This would look like MR 2,3. This is equivalent to 2^2. Then I would set the even register as 2 again since the result is stored across the even-odd register but the even register is just either F's for negative or 0's for positive. This would look like LA 2,2. Then I would multiply the result by the second register by doing MR 2,2. This would be equivalent to 2^3. I would repeat this process as many times as desired to go up to whatever power of 2 I need.
2. One way to determine if the divisor used in a modulus computation is negative is to look at the signs of the remainder and quotient. If the remainder and quotient are opposite signs, that means the divisor was negative. My method for this would be as follows (assuming registers 2 and 3 are the even-odd pair):

```
…                       DIVISION HAPPENED HERE
        LA 4,0          LOAD 0 INTO REG 4
        CR 2,4          CHECK SIGN OF REMAINDER
        BC B'0100',NREM REMAINDER IS NEGATIVE
        BC B'0010',NPOS REMAINDER IS POSITIVE
        B   END         REMAINDER = 0
…
NREM  DS 0H                     NEGATIVE REMAINDER
        CR 3,4          CHECK SIGN OF QUOTIENT
        BC B'0010',DNEG QUOTIENT IS POSITIVE (OPPOSITE SIGN), MOVE
        B  END                  QUOTIENT IS NOT OPPOSITE SIGN
…
NPOS  DS 0H             POSITIVE REMAINDER
        CR 3,4          CHECK SIGN OF QUOTIENT
        BC B'0100',DNEG QUOTIENT IS NEGATIVE (OPPOSITE SIGN), MOVE
        B  END                  QUOTIENT IS NOT OPPOSITE SIGN
…
DNEG   DS 0H                    THE DIVISOR WAS NEGATIVE
…                       DO STUFF HERE FOR NEGATIVE DIVISOR
        B END           MOVE TO END OF IF STATEMENT
END     DS 0H                   PLACE TO PICK UP FROM AFTER
STATEMENTS
…
```

3. My code skeleton:

```
        LA 2,x
        LA 3,y
        CR 2,3          COMPARE X AND Y
        BZ ZERO         BRANCH ON CC 0 (EQUAL)
```

```
        BC B'0100',LESS     BRANCH ON CC 1 (LESS THAN)
…                           X > Y
        B   CONT            MOVE TO END
ZERO    DS 0H                       X = Y
…
        B   CONT            MOVE TO END
LESS    DS 0H               X < Y
…
CONT    DS 0H                       PLACE TO PICK UP FROM AFTER
STATEMENTS
…
```