MULTIPLICACIÓN

; Inicialización

ld R1, num1 ; Cargar num1 en R1

ld R2, num2 ; Cargar num2 en R2 (contador de multiplicación)

ldi R3, 0 ; Inicializar el acumulador (resultado) a 0 en R3

; Bucle de suma repetitiva

inicio\_bucle:

add R3, R1, R3 ; Sumar R1 (num1) al acumulador (R3)

sub R2, R2, 1 ; Decrementar el contador (R2)

testz R2 ; Verificar si el contador (R2) es cero

jnc inicio\_bucle ; Saltar al inicio\_bucle si R2 no es cero

; Almacenar el resultado

st resultado, R3 ; Guardar el valor de R3 (resultado) en memoria

; Fin del programa

DIVISIÓN

; Inicialización

ld R1, dividendo ; Carga el dividendo en R1

ld R2, divisor ; Carga el divisor en R2

ldi R3, 0 ; Inicializa el cociente en 0 en R3

; Verificar si el divisor es cero

testz R2 ; Verifica si el divisor es cero

jc division\_cero ; Si el divisor es cero, saltar a manejo de error

; Bucle de restas repetitivas

inicio\_bucle:

cmp R1, R2 ; Compara el dividendo (R1) con el divisor (R2)

jc fin\_bucle ; Si R1 es menor que R2, salta al fin del bucle

sub R1, R1, R2 ; Resta: R1 = R1 - R2

add R3, R3, 1 ; Incrementar el cociente en 1

jmp inicio\_bucle ; Repetir el bucle

fin\_bucle:

; Guarda el cociente (R3) y el residuo (R1)

st cociente, R3 ; Guardar el cociente en memoria

st residuo, R1 ; Guardar el residuo en memoria

jmp fin\_programa ; Terminar programa

division\_cero:

; Manejar error de división por cero

fin\_programa:

; Fin del programa