

Binary representation for operations

```
# Assume $0 is always 0
# Initial Setup
00001100      ; li $v0, 4 (load immediate for syscall to print string)
00001101      ; li $v0, 5 (load immediate for syscall to read integer)
00001110      ; li $v0, 1 (load immediate for syscall to print integer)
00001111      ; li $v0, 10 (load immediate for syscall to exit)

00010000      ; la (load address for string prompts, binary representation is illustrative)

# Move Operations
00010001      ; move (move data between registers, binary representation is illustrative)

# Arithmetic Operations
01111011      ; add $3, $1, $2 # Add $1 and $2, result in $3
10011011      ; sub $4, $1, $2 # Subtract $2 from $1, result in $4
10111011      ; mul $5, $1, $2 # Multiply $1 and $2, result in $5
11011011      ; div $6, $1, $2 # Divide $1 by $2, result in $6

# Conditional Branch
01111000      ; bgt $3, $4      # If addition result > subtraction result, branch

# Branch if Equal
00010010      ; beq (branch if equal, used for conditional logic in the program)

# Jump
00010011      ; j (unconditional jump, used to return to the start or exit)

# Print Results
01101011      ; pr $3          # Print addition result
10001011      ; pr $4          # Print subtraction result
10101011      ; pr $5          # Print multiplication result
11001011      ; pr $6          # Print division result
```