



```
////////////////////////////////////
////////////////////////////////////
// JUMP TO MAIN (Starts at Instruction Memory Address 0x0)
// Your code will start after the Validation code.
// The program must jump over the Validation code to start at the
appropriate
// address.
// DO NOT CHANGE THE CODE IN THIS SECTION OF THE PROGRAM.

LDI R0, 6
SHL R0, R0
JMP R0

////////////////////////////////////
////////////////////////////////////
// VALIDATION CODE (Starts at Instruction Memory Address 0x3)
// This section:
// - Loads the value of Data Memory[8] into R1
// - Loads the value of Data Memory[9] into R2
// - Loads the value of Data Memory[10] into R3
// The program halts after doing this.
// DO NOT CHANGE THE CODE IN THIS SECTION OF THE PROGRAM.

LDI R0, 7
INC R0, R0
LD R1, R0
INC R0, R0
LD R2, R0
INC R0, R0
LD R3, R0
LDI R0, 0
BRZ R0, 0

////////////////////////////////////
////////////////////////////////////
// YOUR CODE (Starts at Instruction Memory Address 0xC)
// Insert your code starting here, between the VALIDATION CODE and the
JUMP TO
// VALIDATION.
// DO NOT MAKE THE LAST LINE OF YOUR CODE A HALT.

LDI R7,0 // start memory counter in 0
LD R1,R7 // load first number from the array as the initial max
LD R2,R7 // load first number from the array as the initial min
LD R3,R7 // initialize total sum to first number in array
INC R7,R7 // increment mem counter
LDI R6,7 // load 7 in R6
INC R6,R6 // increment R6 to 8 to use R6 for loop comparison
// loop:
LD R0,R7 // load a number from the array
ADD R3,R3,R0 // add loaded number to total sum
SUB R4,R0,R1 // compare loaded number with maximum using a subtraction
BRN R4,2 // if it's negative, skip next instruction
ADI R1,R0,0 // else R0 is the new maximum, save R0 in R1
```



```
SUB R4,R2,R0 // compare loaded number with minimum using a subtraction
BRN R4,2 // if it's negative, skip next instruction
ADI R2,R0,0 // else R0 is the new minimum, save R0 in R3

INC R7,R7 // increment mem counter
SUB R0,R7,R6 // compare counter with 8
BRN R0,-10 // if it's lower than 8, repeat

SHR R3,R3 // shift once to divide sum by 2
SHR R3,R3 // shift twice to divide sum by 4
SHR R3,R3 // shift three times to divide sum by 8, R3 now is the
average

ST R6,R1 // store maximum in memory location 8
INC R6,R6 // increment address, is now 9
ST R6,R2 // save minimum in memory location 9
INC R6,R6 // increment address, is now 10
ST R6,R3 // save average in memory location 10

////////////////////////////////////
////////////////////////////////////
// JUMP TO VALIDATION
// The last line of your code should precede this section.
// DO NOT MAKE THE LAST LINE OF YOUR CODE A HALT.
// This section of the code jumps back to the Validation section.
// DO NOT CHANGE THE CODE IN THIS SECTION OF THE PROGRAM.

LDI R0, 3 // Use R0 to point to the validation routine.
JMP R0 // Jump to the validation routine.
```