

Due: **Friday, October 19 at 9:00 am.**

Submission: Please hand in a (tidy) paper copy at the beginning of class.

1. Translate the C code segment below into AVR assembly language. Assume that variables  $n$  and  $sum$  have already been declared as signed 8-bit numbers, and that space has been allocated for them in memory locations 0x0064 and 0x0065, respectively. Your program segment should read  $n$  from memory, perform the computations, and store the sum back to memory.

```
sum = 0;
for (i = 1; i < n; i++) {
    sum += i;
}
```

2. Translate the C code segment below into AVR assembly language. Assume the variables are all 8-bit signed numbers and that they have been assigned to sequential memory locations (in alphabetical order), starting with 0x0060. Your program segment should include statements to read the necessary values from memory and store the results back to memory.

```
while (n < 20) {
    m--;
    if (n >= m) {
        n = n + 2;
    }
    else {
        n++;
    }
}
```

3. Translate the C code segment below into AVR assembly language. Make the same assumptions, and use the same instructions, as in question 2.

*Hint: Think about how you might transform the C code to remove the compound condition.*

```
if (a >= 5 && a < 15)
    b = c;
else
    b = d;
```

4. Translate the C code segment below into AVR assembly language. Make the same assumptions, and use the same instructions, as in question 2.

Notice that the conditional operator is *greater than*, not greater than or equal to. The AVR assembly language does not have an instruction to test for greater than; however, you can write this code with only the instructions we learned in class.

```
if (a > b) {
    a = 0;
}
else {
    a--;
}
```