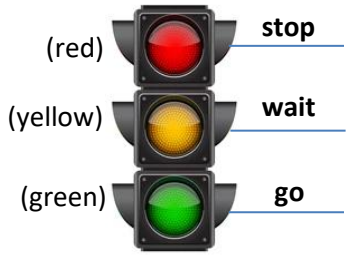
Mr. Bellavia

ASSIGNMENT 02 - COMPUTER STORAGE & PROCESSING

1. Calculate the size in bytes of your full name (refer to question 1 from the last exercise where you converted your full name to binary).

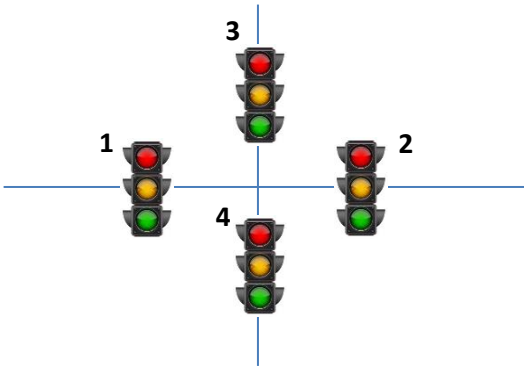
Answer (show your calculation):

2. Traffic lights are hard coded to work at intersections to control the flow of traffic:



- Along with the basic functionality of stop, wait, and go, traffic lights are also timed appropriately where each light stays on for a certain amount of time.

Let's assume there are two sets of traffic lights perpendicular to each other at an intersection:



Traffic lights **1 & 2** are synced to do the same thing since they are controlling the horizontal traffic of this intersection diagram. Traffic lights **3 & 4** are synced in the same manner controlling the vertical traffic.

To control the flow of traffic, there exists a computer program that controls these traffic lights so that they work together (i.e., they are synchronized). Consider the following table of instructions that may exist for such a program:

Memory Address	Machine Instruction	Action
0	00100001	Turn on red light for light 1
1	00000101	Turn off red light for light 1
2	00000001	Turn on red light for light 2
.	.	.
.	.	.
7	11100000	Wait 5 seconds
8	10101011	Wait 30 seconds

.	.	.
.	.	.

Note: A **Memory Address** always start at 0 and increases by 1. A **Machine Instruction** is just a unique and arbitrary set of **8 bits** made up of **0s & 1s**. There is **no special order** when listing your instructions.

a) Create your own table of instructions for this traffic light intersection program (you may use the instructions given in the above table to get started):

Memory Address	Machine Instruction	Action

Note: If you need to add more rows to the above table: right click on a cell in the table, select **'Insert'** from the popup menu, then click **'Insert Rows Above'** or **'Insert Rows Below'** appropriately.

b) Now that you have all your instructions, create a list of memory address calls in an order that would make this intersection functional. Assume that this list of memory addresses will be called repeated by your program so that the flow of traffic never stops.

Your answer should look something like this: **3, 4, 5, 3 . . .** (where each number is a **memory address** from the above table)

Answer: