# Exercise 1:

The first issue I see with the example code is use of a mystery number “20”. Using the explicit number assumes the array will only contain 20 elements. If the array does not contain that amount, either too many or too little, we can run into some errors. The second issue I see is the initialization of “i =1”. This will make the loop we have created always skip the first element in the array. Arrays start at the zeroth index not the first unless explicitly allocated as such. My modifications are below to correct the above issues.

int values[20];  
for (int i = 0; i <= sizeof(values)/sizeof(values[0]); i++)  
{  
   values[i] = i + i;  
}

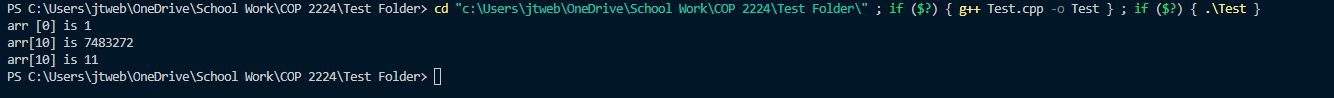
# Exercise 2:

An index is a location in memory allocated for a value that is specified when an array is initialized and instantiated. Think of the array as a bookshelf and an index the shelfs on the bookshelf. The index specifies where the book is stored on the bookshelf.

Legal values for an array of size 9 are from 0-8. Arrays start any the zeroth index so you must remember to take the size of the array and minus 1 to get the appropriate locations.

# Exercise 3:

Unfortunately, I was unable to generate an error message as C/C++ does not generate bounds errors by default. I was able gather an example of what happens when accessing memory that has not been allocated a value. Hopefully this will suffice.



# Exercise 4:

Arrays and Vectors are the same:

They both store data at specified memory addresses that can be accessed later.

They both store a specified data type.

Arrays and Vectors are different:

A Vector is a sequential-based container whereas an array is a data structure that stores a fixed number of elements.

Arrays can be implemented in a static or dynamic way whereas vectors can only be implemented dynamically.

Real-life situation when an Array would be preferred:

A real-world example would be a DMV. Arrays are better suited for frequent access of elements and verifying information such as address, date of birth, social security number, and so on would be READ ONLY meaning no editing would be implemented.

A real-life situation when a Vector would be preferred:

A real-world example would be a hospital. Vectors are better at data insertion and deletion as opposed to an array. Hospitals constantly are updating information pertaining to patient health and treatment. A vector would be perfect due to the amount of data manipulation needed/