Arrays
https://www.cplusplus.com/files/tutorial.pdf Pages (54-59)

Arrays are an advanced variable type that allows you to put multiple values into a single variable name, like a list. Arrays are declared in a slightly different way to normal variables.

```
int test[10];
```

We still have to give the variable a type and a name, but we also have to add a size (in this case, the size is 10). This variable "test" is now ten different integers, placed in a group. To get a single value we need to index the array, or grab from a specific location in the array. To index the array, you refer to the name of the variable, plus the location index in brackets, like this:

```
test[4];
     /\
```

This refers to the 5th integer in test.

ARRAYS INDEX FROM 0. THE FIRST ARRAY POSITION IS ALWAYS [0].

That line has been bolded to make it easier to remember, because arrays can be tricky. Arrays are only as big as you make them, and if you go off the end of an array you will crash your program and cause a `Segmentation Fault`. This is not the end of the world, and it does not damage your computer, but it can be very hard to figure out where and why it happened, because your compiler will not give you much info. This can be even harder to fix than the normal errors we get, so BE CAREFUL.

Generally a while loop is used to access every value in an array. This example code uses two while loops to first fill an array with the numbers 0 to 9, and then print them in that order.

```
int test[10];
int i = 0;
while(i < 10){
    test[i] = i;
    i++;
}
i = 0;
while(i < 10){
    cout << test[i] << endl;
    i++;
}
```

Notice that we start from zero and go to nine, not ten. This is, once again, because ***ARRAYS INDEX FROM 0.*** Please do not forget that. One of the nice tricks that comes from indexing from 0 is that we can use `while(i<10)` and not `while(i<=10)`, because we want to go from 0 to one less than the size of the array.

You can use arrays to store lots of user values to do math on, like this:

```
int test[100];
int i = 0;
while(cin >> test[i]){
    if(test[i] == 0){break;}
    i++;
}
```

This gives us a `test` array with a number in every location from "0" to "i".