Arrays and functions

- Arrays can be passed as parameters to functions
  - Formal array parameters must have an empty []

```
void myArrayFunction( int one, int many[] );
```

  - An array parameter is used just like an array local variable
  - But, arrays are implicitly passed by ***reference***!
    - All changes to an array parameter change memory outside the function scope <=> ***the values of the original array will also be changes.***
    - You never would have a formal parameter `int &many[]`

value	15	10	10	10	10	10	10
index	0	1	2	3	4	5	6

Array pass the memory/address/reference of its first element.

Arrays and functions

- When ***calling a function*** with an array parameter:
  - You must provide ***an array*** variable for that parameter
  - It will be passed in by ***reference***
  - Syntax: just the variable name, like any other parameter
    - Note the difference syntax for the formal parameter vs. the actual parameter

```
int my_function( int param[] );
…

int x, array_var[20];
x = my_function(array_var);
```

Arrays and functions

- Four different array syntaxes
  - Declaration
    - `int y[50];`
    - Subscript notation to specify the size of the array
  - Accessing an element
    - `y[20] = 10; // index 20 is the 21st element in the array`
    - Subscript notation to specify the *index*
  - Formal parameter
    - `y[]`
    - Must tell the function that the parameter is an array (not a regular variable)
  - Actual parameter
    - `y;` No additional information necessary