# Copy Array:

copy()

input:

int A[n]

takes in array of integers, A

output:

outputs a copy/duplicate of array A, this duplicate is B, array of integers

methods:

for loop

```
for(i=0; i<n; ii){
    duplicate array
    assign to variable

}
```

## Sum of All Elements:

addAll()

input:

int A[n]

A, an array of integers

output:

sum, the sum of the n elements in the array, A

methods:

```
for loop
    for (int i=0 , i<n)
        sum = sum + A
```

# Sum of Two Arrays:

## add Arrays ( ) :

input: 2 separate arrays w/ # values

int ([n] arr)

A array of n numbers

int ([n] arr)

B array of n numbers

Output:

C array adds together the numbers in A position i and B position i

outputs the sum of A position i and B position i in new array, C, in position i

method:

```
For loop
    For i = 0 to n {
        $A_n \pm B_n = C_n$
    returns $C_n$ }
```

# Product of All Elements:

multiplyAll()

input:

int[n]

A, an array of integers up to n

output:

Product, the product of multiplying n elements, in A, together

methods:

```
For loop
    For (i ← 0 ; i < n)
        Product = A[i] · A
```

# Find Average of Elements:

FindAverage()

input:

int [n] A

A, an array of integers

output:

Average, the average of all n values in A

methods:

```
for loop
    for (i=0; i<n; i++)
        Avg = A[i] + A[i]
```

# Swap Elements:

swap()

input:

int A[n]

A, an array of n integers

output:

returns A, but with n integers swapping places

methods:

for loop

```
for(i=0; i<n)
    if i = y (index) || i = z (index)
        switch places
```

## Find Elements:

isElement()

input:

int A[n]

A, an array of integers

y, an integer value

output:

a boolean value of true or false depending on if y is in A or not

methods:

```
For loop
    for (i=0; ...)
        if A.contains(y)
            return Boolean true
        else if:
            return Boolean false
```

## Find Index of Element:

indexOf()

input:

int A[n]

A, an array of n integers

y, an integer value

output:

the index of y if it is in A, otherwise the value -1

methods:

For loop

```
For(i=0. . .)
    if A.contains(y)
        return (index of y)
    else if
        return (-1)
```

# Find Min:

FindMin( )

input:

int A[n]

A, an array of n integers

output:

returns the smallest integer value(s) in the array, A

methods:

For loop

```
min = 5

for loop (i≤0, i<n) {
 if y => min,
   min stays same
   else: y < min
     min = v
```

## Find Index of Min:

1

findMin index()

input:

int A[n]

A, array of n integers

output:

returns the index value of the minimum value in A

methods:

for loop

```
min = y (smallest int)
for (.. ..)
    find min
    return index
```

# Find Max

FindMax( )

input:

int A[n]

~~A~~, an integer array

output:

the max/largest value of A

methods:

```
for loop
    max= y (largest)
    For(....)
        if (max => y)
            return max
        else:
            max== ~~y~~ integer
```

## Find Index of Max:

Find Max Index()

input:

int A[n]

A, an array of n integers

output:

the index of the max valve in the array

Methods:

For loop

```
for(. . . .)
min = y
Find min
return index
```

## Reverse an Array:

reverse()

input:

int A[n]

A, array of A integers

output:

the reversed values of A

methods:

For loop

```
For (i=max value, i--)
    (counts down from Max value
```

# Intersection of Two Arrays

who is in both arrays

intersection()

input:

int A[n]

takes in Array A of n numbers

int B[n]

takes in Array B of n numbers

output:

C array contains any duplicate numbers, n, that are in both array A and B

methods:

for loop

```
for (i = 0 to n){
    if $A_n == B_n$;
        return $C_n$;
}
```

# Union of Two Arrays.

union()

Input:

int A[n]

A, an array of n numbers

int B[n]

B, an array of n numbers

output:

C, a new array that contains the n numbers from array A & B so, two arrays combined

Methods:

for loop

```
for (i=0 to n){
    C = A+B
    return C
    }
```