Function : Copy( )

Input : array A (of integers)

Output : newArray (of integers)

Description:

```
function copy ( array A )   input
    newArray = new ArrayList <int>
    for (i=1 to n)   size of array A (length)
        newArray[i] = A[i]
    return newArray ← output (new array list of copied A)
```

Function : addAll( )

Input : array A (of integers)

Output : Sum $A_1 + A_2 + \cdots + A_n$

Description:

```
function addAll ( array A )   input
    sum=0
    for (i=1 to n)   size of array A (length)
        sum = sum + A[i]
    return sum ← output
```

Function : addArrays( )

Input : array A (of integers) , array B (of integers)

Output : sumArray $A_1 + B_1$, $A_2 + B_2$, $A_3 + B_3$, ... , $A_n + B_n$

Description:

```
function addArrays ( array A . array B) ← input 2 array.
    sumArray = new ArrayList <int> ← create new array that stores the added terms
    for (i=1 to n) ← array size
        sumArray [i] = A[i] + B[i]
    return sumArray ← output
```

Function : multiplyAll( )

Input : array A (of integers)

Output : product $A_1 \cdot A_2 \cdot A_3 \cdot \cdots \cdot A_n$

Description:

```
function multiplyAll ( array A) ← input
    product = 1
    for (i=1 to n) ← size of array A (length)
        product = product * A[i]
    return product ← output
```

Function : findAverage()

Input : array A (of integers)

Output : average

Description:

```
function findAverage (array A) ← input

    total = addAll (array A)

    average = total / A.size() ← get size of A (n)


    return average ← output
```

Function : swap()

Input : array A (of integers) , a (integer) , b (integer)

Output : array A

Description:

```
function swap( array A, int a. int b)

        swap1 = A[a]

        A[a] = A[b]        → swap the order in A.

        A[b] = swap1

        return A
```

Function : isElement( )

Input : array A (of integers) , a (integer)

Output : Boolean

Description:

```
function isElement (array A , int a)
    for (i = 1 to n)
        if A[i] == a
            return true
    return false
```

Function : Indexof( )

Input : array A (of integers) , a (integer)

Output : index k

Description:

```
function indexOf (array A , int a)
    for (i = 1 to n )
        if A[i] == a
            return i
    return -1
```

Function : findMin()

Input : array A (of integers)

Output : smallest Integer a

Description:

```
function findMin( array A )
    a = A[1]  ← first term in A
    for (i=2 to n)
        if A[i] < a
            a = A[i]
    return a
```

Function : findMinIndex()

Input : array A (of integers)

Output : Index k of smallest integer a

Description:

```
function findMinIndex( array A )
    k = 1   (first integer index in A)
    for (i=2 to n)
        if A[i] < A[k]
            k = i
    return k
```

Function : findMax()

Input : array A (of integers)

Output : largest integer a

Description:

```
function findMax( array A )
        a = A[1] ← first term in A
        for (i=2 to n)
            if A[i] > a
                a = A[i]
        return a
```

Function : findMaxIndex()

Input : array A (of integers)

Output : index k of largest integer a

Description:

```
function findMinIndex ( array A )
        k = 1    (first integer index in A)
        for (i=2 to n)
            if A[i] > A[k]
                k = i
        return k
```

Function : reverse()

Input : array A (of integers)

Output : reversed array B (of integers)

Description:

```
function reverse (array A)
        B = new Array List <int>
        for (i=1 to n)
            B[i] = A[n-i]
                              ↑ Start from the last element
      return B
```

Function : intersection()

Input : array A (of integers), array B (of integers)

Output : new array C of same elements in A and B. (of integers)

Description:

```
function intersection (array A, array B)
        Array C = new ArrayList <int>
        for (i=1 to n)
            if isElement (B, A[i])
                C.add (A[i])
        return C
```

Function : union ()

Input : array A (of integers) , array B (of integers)

Output : new array C (of integers)

Description:

```
function union (array A, array B)
        array C = new ArrayList <int>
        for (i=1 to n)
            C. add(A[i])
        for (i=1 to n)
            for (k=1 to m)
                if B[i] != A[k]
                    C. add (B[i])
    return C
```

Function : sort ()

Input : array A (of integers)

Output : array B (of sorted integers)

Description:

```
function sort (array A)
    for (i=1 to n)
        min = findMin (A)
        B. add (min)
```

```
            A. remove (min)
        return B
```