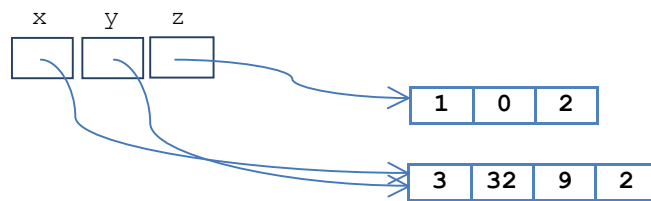


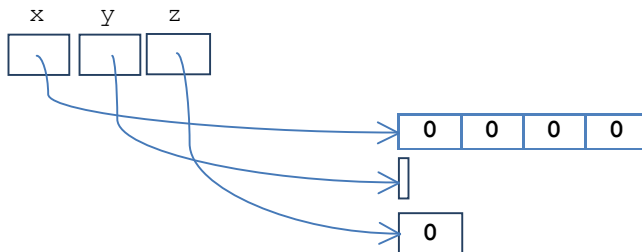
Practice Problems: Array Problems

1. Understanding code

```
public class Array-Assignment {  
    public static void main(String [] args) {  
        int [] x = new int[3];  
        int [] y = {3, 5, 9, 2};  
        x[2] = y[3];  
        x[0]++;  
        y[1] += y[2] * y[0];  
        int [] z = x;  
        x = y;  
    }  
}
```



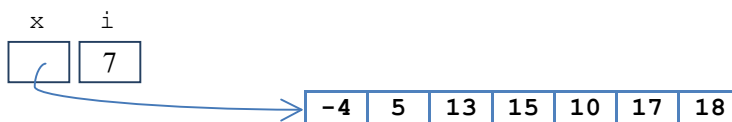
```
public class Array-Length {  
    public static void main(String [] args) {  
        int [] x = new int[4];  
        int [] y = {};  
        int [] z = {0};  
        System.out.println("x has " + x.length + " elements");  
        System.out.println("y has " + y.length + " elements");  
        System.out.println("z has " + z.length + " elements");  
    }  
}
```



screen

```
x has 4 elements  
y has 0 elements  
z has 1 elements
```

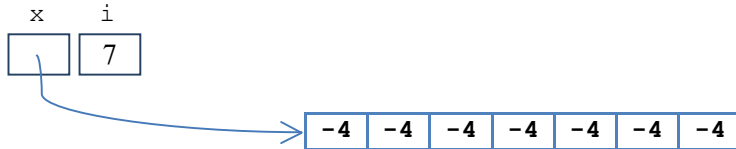
```
public class Array-With-Loop1 {  
    public static void main(String [] args) {  
        int [] x = {-4, 9, 8, 2, -5, 7, 1};  
        for(int i=1; i<x.length; i++) {  
            x[i] += x[i-1]; // notice: += instead of =  
        }  
    }  
}
```



```

public class Array-With-Loop2 {
    public static void main(String [] args) {
        int [] x = {-4, 9, 8, 2, -5, 7, 1};
        for(int i=1; i<x.length; i++) {
            x[i] = x[i-1];
        }
    }
}

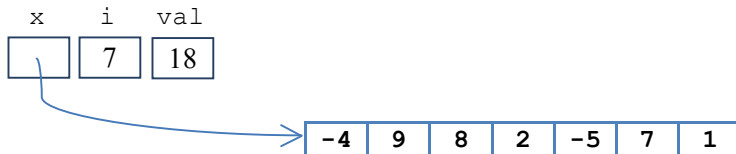
```



```

public class Array-With-Loop3 {
    public static void main(String [] args) {
        int [] x = {-4, 9, 8, 2, -5, 7, 1};
        int val = 0;
        for(int i=0; i<x.length; i++) {
            val = val + x[i];
        }
    }
}

```



2. Writing Java Programs with Arrays

Write a counter controlled loop to solve the following problems. Each one will involve an array.

1. Read in 500 ints from the keyboard, and store them in an array. Find the maximum and minimum values in the array, and display them on the screen.

```

Scanner kb = new Scanner(System.in);

int [] intArr = new int[500];
for(int i=0; i<intArr.length; i++)
    intArr[i] = kb.nextInt();

// Accumulate algorithm to find maximum element
int max = intArr[0];
for(int i=1; i<intArr.length; i++) {
    if(intArr[i] > max) {
        max = intArr[i];
    }
}
System.out.println("max = " + max);

// Accumulate algorithm to find minimum element
int min = intArr[0];
for(int i=1; i<intArr.length; i++) {
    if(intArr[i] < min) {
        min = intArr[i];
    }
}
System.out.println("min = " + min);

```

2. Read in 500 ints from the keyboard, and store them in an array. Find the *position* (or *index*) of the maximum and minimum values in the array, and *swap* them (move the biggest element to the position of the smallest, and move the smallest element to the position of the biggest).

```
Scanner kb = new Scanner(System.in);
int [] intArr = new int[500];
for(int i=0; i<intArr.length; i++)
    intArr[i] = kb.nextInt();

// Accumulate algorithm to find position of max element
int max = intArr[0];
int maxPos = 0; // maxPos starts at position 0
for(int i=1; i<intArr.length; i++) {
    if(intArr[i] > max) {
        max = intArr[i];
        maxPos = i;
    }
}

// Accumulate algorithm to find position of min element.
int min = intArr[0];
int minPos = 0;
for(int i=1; i<intArr.length; i++) {
    if(intArr[i] < min) {
        min = intArr[i];
        minPos = i;
    }
}

// swap the biggest and smallest elements
intArr[minPos] = max;
intArr[maxPos] = min;

// here's a handy command to print out the entire array
System.out.println(Arrays.toString(intArr));
```

3. Read 500 ints from the keyboard, and store them in an array. Display "true" on the screen if there is an even number of even numbers among these 500. Otherwise, display "false".

```
Scanner kb = new Scanner(System.in);
int [] intArr = new int[500];
for(int i=0; i<intArr.length; i++) {
    intArr[i] = kb.nextInt();
}

// Accumulate algorithm to find number of even elements
int countEven = 0;
for(int i=0; i<intArr.length; i++) {
    if(intArr[i] % 2 == 0) {
        countEven++;
    }
}

if(countEven % 2 == 0) { // even number of evens
    System.out.println("true");
} else {
    System.out.println("false");
}
```