Mr. Fahrenbacher
AP Computer Science A
Array Operations

Name: _____

Operation: Fill an array with values

Example: Fill an array with the values 1 through the length

```
int[] myNumbers = new int[50];

for(int i=0; i<myNumbers.length; i++) //it's better to use .length instead of 50... why?
{
        myNumbers[i] = i+1;
}
```

1) Create an array of size 25 to hold integers. Store in the array the values 0 through 24 in **reverse** order.

2) Create an array of size 10 to hold integers. Store in the array 10 values read in from the user.

//e.g. Prompter.promptForInteger("Give me a number!");

3) Create an array of size 30 to hold integers. Store in the array 30 values chosen randomly between 2 and 10.

Operation: Print out values in the array

Ex: Print out all the numbers in the array

```
int[] myArray = ....; //I'm intentionally not giving you the length of the array - do you know why?

for(int i=0; i<myArray.length; i++)
{
        System.out.print(myArray[i] + " "); //I put a space between each printed number
}
System.out.println(); //I'm adding a new line afterwards
```

4) Print out all the numbers in the array that are even.

5) Print out all the numbers in the array that are at an even **index**.

6) Print out all the values in the array that are less than the value after them in the array.

Operation: Perform a calculation involving all the elements in the array

Ex: Sum (add) up all the numbers in the array

```
int[] theData = ....; //you can't see the length or what's in the array!
int sum = 0;

for(int i=0; i< theData.length; i++)
{
        sum += theData[i];
}
System.out.println(sum); //I'm printing the info - you might want to return this data in a method
```

7) Calculate the sum of all numbers in the array that are odd

8) Calculate the **average** of all numbers in the array that are at odd **indexes**.

9) Calculate which number is the biggest in the array

10) Calculate at which index the biggest number in the array is at (if multiple, pick the first)

11) CHALLENGE: FIll an array with the first 50 Fibonacci numbers (0, 1, 1, 2, 3, 5, 8, ...).