Exercise – Partially filled arrays

1. Create an array called *sample* that holds the following array of integers:

54	19	59	98	85	71	86	45	50

```
int [] sample = {54,19,59,98,85,71,86,45,50};
```

2. Practice: Write a segment of code that will rearrange the array in reverse order. That is, *sample*[0] will swap values with *sample*[8] and so on.

```
int [] sample = {54,19,59,98,85,71,86,45,50};
      int [] reverse = new int [sample.length];
      int count = 0;
 for (int x = sample.length - 1; x >= 0; x--) {
            reverse[count] = sample[x];
            count++;
      }

 sample = reverse;
      for (int x = 0; x < sample.length; x++) {
            System.out.print(sample[x] + "\t");
      }
```

3. Practice: Create an array called *sample2* that is the same size as *sample* and copy all elements of *sample* to *sample2* one elements at a time.

```
int [] sample = {54,19,59,98,85,71,86,45,50};
      int [] sample2 = new int [sample.length];

      for (int x =0; x < sample.length; x++) {
            sample2 [x] = sample[x];
      }
```

4. Create an array called *sample3* that is one index larger than *sample* and copy all elements of *sample* to *sample2* one elements at a time. Leave the last index empty. Print all the elements of *sample3*.

```
int [] sample = {54,19,59,98,85,71,86,45,50};
      int [] sample3 = new int [sample.length + 1];

      for (int x =0; x < sample.length; x++) {
            sample3 [x] = sample[x];
      }

      for (int x =0; x < sample3.length; x++) {
            System.out.print(sample3[x] + "\t");
      }
```

5. Create a method called *increasefirst* that will receive an array as a parameter. This method will create a local array that is one index larger than the parameter sent in. This method will copy the elements to the new array in order while leaving the first index empty and return that array to the main program.

	54	19	59	98	85	71	86	45	50

```
            int [] array = {54,19,59,98,85,71,86,45,50};
            array = increasefirst(array);
            for (int x = 0; x < array.length; x++)
                  System.out.print(array[x] + "\t");

      }

      public static int [] increasefirst (int [] array) {
            int [] temp = new int [array.length+1];
            int count = 0;
            for (int x = 1; x < temp.length; x++) {
                  temp[x] = array[count];
                  count++;
            }
            return temp;
```

6. Create a method called *increaselast* that will receive an array as a parameter. This method will create a local array that is one index larger than the parameter sent in. This method will copy the elements to the new array in order while leaving the last index empty and return that array to the main program.

54	19	59	98	85	71	86	45	50	

```
            int [] array = {54,19,59,98,85,71,86,45,50};
            array = increaselast(array);
            for (int x = 0; x < array.length; x++)
                  System.out.print(array[x] + "\t");

      }

      public static int [] increaselast (int [] array) {
            int [] temp = new int [array.length+1];
            for (int x = 0; x < array.length; x++)
                  temp[x] = array[x];

            return temp;
      }
```

7. Create a method called *removefirst* that will receive an array as a parameter. This method will create a local array that is one index smaller than the parameter sent in. This method will copy the elements to the new array in order except the first element and return that array to the main program.

19	59	98	85	71	86	45	50

```
public static void main(String[] args) {
		// TODO Auto-generated method stub
		int [] array = {54,19,59,98,85,71,86,45,50};
		array = removefirst(array);
		for (int x = 0; x < array.length; x++)
			System.out.print(array[x] + "\t");

	}

	public static int [] removefirst (int [] array) {
		int [] temp = new int [array.length-1];
		for (int count = 0, x =0; x < temp.length; x++,count++)
			temp[x] = array[count];
		return temp;
	}
```

8. Create a method called *removelast* that will receive an array as a parameter. This method will create a local array that is one index smaller than the parameter sent in. This method will copy the elements to the new array in order except the last element and return that array to the main program.

54	19	59	98	85	71	86	45

```
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int [] array = {54,19,59,98,85,71,86,45,50};
		array = removelast(array);
		for (int x = 0; x < array.length; x++)
		System.out.print(array[x] + "\t");
	}

	public static int [] removelast (int [] array) {
		int [] temp = new int [array.length-1];
		for (int count = 0, x = 0; x < temp.length; x++,count++)
			temp[x] = array[count];
		return temp;
	}
```