

1) Finish this method that fills an array with these values: 1, 2, 3..., n-1, n, n-1, ..., 3, 2, 1

//Hint: you need to make an array (how large?) Try an example with n = 5 to get the proper motivation...

```
public int[] createWedge(int n) {  
  
    int[] wedge = new int[2*n - 1];  
    int val = 1;  
    for(int i = 0; i < wedge.length; i++)  
    {  
        wedge[i] = val;  
        if(i < n)  
            val++;  
        else  
            val--;  
    }  
    return wedge;  
}
```

```
int[] wedge = new int[2*n - 1];  
for(int i = 0; i <= wedge.length/2; i++)  
{  
    wedge[i] = i + 1;  
    wedge[wedge.length-1-i] = i + 1;  
}  
return wedge;
```

2) If you take any two positive integers m and n ($m > n$), then new numbers a, b, and c can be calculated where:

$$a = m^2 - n^2 \qquad b = 2mn \qquad c = m^2 + n^2$$

These numbers a, b, and c form a pythagorean triple:

$$a^2 + b^2 = c^2$$

Create a method that takes m and n as parameters and returns an array of 3 elements containing the values of a, b, and c. Write a sentence or two afterwards describing special cases for values of m and n that should be considered.

```
public int[] triples(int m, int b)  
{  
    int[] array = new int[3];  
    array[0] = m*m - n*n;  
    array[1] = 2*m*n;  
    array[2] = m*m + n*n;  
    return array;  
}
```