Algorithms
GROUP WORK

Warmups

- 1 person use Zoom to screenshare and solve the problems together
- Submit your .js file to Mattermost at the end.

```
// Build a function that prints the numbers
// from 1 to 50

// Build a function that prints the numbers
// from 1 to a given number
```

Medium Heat

```
// Build a function that prints the numbers
// from 10 to 1 and also prints whether the
// number is even or odd
// 10 even
// 9 odd
// 8 even...
```

FizzBuzz

```
// Build a function that goes from 1 to a
// given number and prints the following...
// "Fizz" if the number is divisible by 3
// "Buzz" if the number is divisible by 5
// "FizzBuzz" if the number is divisible by both
// otherwise print just the number itself
```

Advanced Ninjas (optional)

Array: Rotate

Implement **rotateArr(arr, shiftBy)** that accepts array and offset. Shift **arr** 's values *to th e right* by that amount. 'Wrap-around' any values that shift off array's end to the other side, s o that no data is lost. Operate in-place: given **([1,2,3],1)** , change the array to **[3,1,2]** . Don' t use built-in functions . **Second:** allow negative **shiftBy** (shift L, not R) . **Third:** minimize memory usage.

Array: Filter Range

Alan is good at breaking secret codes. One method is to eliminate values that lie within a specific *known* range. Given **arr** and values **min** and **max** , retain only the array values between **min** and **max** . Work in-place: return the array you are given, with values in original order. No built-in array functions.