

1. Given two numbers. Print powers of 2 between that numbers. (without using *Math.pow*).

Input	Output
7, 45	8, 16, 32
0, 150	1, 2, 4, 8, 16, 32, 64, 128

2. A perfect number is a positive integer that is equal to the sum of its proper positive divisors. Print all perfect numbers between 1 and  $n$ .
3. Given a number  $n$  ( $n \geq 0$ ). Print  $n$ th Fibonacci number. (*Fibonacci series: 0, 1, 1, 2, 3, 5, 8, ...,  $a_k = a_{k-1} + a_{k-2}$* ).

Input	Output
2	1
10	55

4. Write a function which will equivalent in === (strict equal ) operator.
5. Given the list of the following readers.

```
[
  { book: "Catcher in the Rye", readStatus: true, percent: 40 },
  { book: "Animal Farm", readStatus: true, percent: 20 },
  { book: "Solaris", readStatus: false, percent: 90 },
  { book: "The Fall", readStatus: true, percent: 50 },
  { book: "White Nights", readStatus: false, percent: 60 },
  { book: "After Dark", readStatus: true, percent: 70 },
]
```

Output the books sorted by the percent in descending order which readStatus is true.

6. A **boomerang** is a V-shaped sequence that is either upright or upside down. Specifically, a boomerang can be defined as a **sub-array of length 3, with the first and last digits being the same and the middle digit being different.**

Create a function that returns the total number of **boomerangs** in an array.

To illustrate:

```
[3, 7, 3, 2, 1, 5, 1, 2, 2, -2, 2]
```

```
// 3 boomerangs in this sequence: [3, 7, 3], [1, 5, 1], [2, -2, 2]
```

### Examples

```
countBoomerangs([9, 5, 9, 5, 1, 1, 1]) → 2
```

```
countBoomerangs([5, 6, 6, 7, 6, 3, 9]) → 1
```

```
countBoomerangs([4, 4, 4, 9, 9, 9, 9]) → 0
```

7. Create a function that takes two dates and returns the number of days between the first and second dates.

### Examples

```
getDays(  
  new Date("June 14, 2019"),  
  new Date("June 20, 2019")  
) → 6
```

```
getDays(  
  new Date("December 29, 2018"),  
  new Date("January 1, 2019")  
) → 3  
  
// Dates may not all be in the same month/year.
```