1. Given an array. Write a recursive function that removes the first element and returns the given array. (without using *arr.unshift(),assign second element to first, third element to second...*)

Input	Output
[6, 78, ‘n’, 0, 1]	[78, ‘n’, 0, 1]
[5]	[]
[]	[]

2. Given an array of nested arrays. Write a recursive function that flattens it. (Hint create function that concats arrays).

Input	Output
[1, [3, 4, [1, 2]], 10]	[1, 3, 4, 1, 2, 10]
[14, [1, [[[3, []]], 1], 0]	[14, 1, 3, 1, 0]

3. Given a number. Write a function that calculates its sum of the digits and if that sum has more than 1 digit find the sum of digits of that number. Repeat that process if needed and return the result.

Input	Output
14	5
29	2
999999999999	9

4. Given an array and a number N. Write a recursive function that rotates an array N places to the left. (*Hint*: to add element to the beginning use *arr.unshift()*)

[‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’] 3	[‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘a’, ‘b’, ‘c’]
[‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’] -2	[‘g’, ‘h’, ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’]

5. Given an object. Invert it (keys become values and values become keys). If there is more than key for that given value create an array.

Input	Output
{ a: ‘1’, b: ‘2’ }	{ 1: ‘a’, 2: ‘b’ }
{ a: ‘1’, b: ‘2’, c: ‘2’ }	{ 1: ‘a’, 2: [‘b’, ‘c’] }
{ a: ‘1’, b: ‘2’, c: ‘2’, d: ‘2’ }	{ 1: ‘a’, 2: [‘b’, ‘c’, ‘d’] }

6. 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.