1. Find the sign of product of three numbers without multiplication operator. Display the specified sign.

Input	Output
-14, 5, 0	“unsigned”
-8, 9, -6	“+”
4, 19, -2	“-”

2. Given a number n (n >= 0). Print n-th Fibonacci number. (Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8 ..., $a_n = a_{n-1} + a_{n-2}$)

Input	Output
0	0
2	1
10	55
20	6765

3. Given an array of a size smaller than 100. It consists of numbers from 0 to 99 in any order. Create a new array where each element from that array is placed under the index of its value. Start from the smallest value and end with the biggest one. If there are missing values, put in its places undefined.

Input	Output
[4, 3, 0, 9]	[0, undefined, undefined, 3, 4, undefined, undefined, undefined, undefined, 9]
[26, 30, 19, 5]	[5, undefined x 13, 19, undefined x 4, 26, undefined x 3]

undefined x 13 - means undefined 13 times.

4. Given an array consisting from the arrays of numbers (like a two-dimensional array). Find sum of each row and print them as an array.

Input	Output
[[3, 4, 5], [1, 0, 0], [4, 5, 4], [8, 8, -1]]	[12, 1, 13, 15]
[[8, 35, 2], [8], [5, 6, -5 , -6], [1, 3, -9, 0,-1]]	[45, 8, 0, -6]
[[1], [2], [3], [4]]	[1, 2, 3, 4]

5. Convert base-2 number to base-10

Input	Output
010	2
10010	18

Input	Output
0101011	43

6. Takes two arrays and insert the second array in the middle of the first array. The first array always has two elements.

Input	Output
[1, 10], [2, 3, 4, 5, 6, 7, 8, 9]	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[15, 150], [45, 75, 35]	[15, 45, 75, 35, 150]
[[1, 2], [5, 6]], [[3, 4]]	[[1, 2], [3, 4], [5, 6]]

7. Enter a positive number. Calculate and print its factorial. (n! = n _ (n-1) _ (n-2) _ ... _ 3 _ 2 _ 1 , 0! = 1)

Input	Output
5	“5! = 120”
1	“1! = 1”
7	“7! = 5040”

8. Write a program to check the validity of password input by users. Validation :

- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [$#@].
- Minimum length 6 characters.
- Maximum length 16 characters.

Input	Output
12asdf	“Invalid”
Aaza1234566#	“Valid”

9. Given a phone number. Write a function to clean it up, so it is valid. The rules are as follows:

- If the phone number is less than 10 and longer than 10 digits assume that it is a bad number
- If the phone number consists of 11 symbols and the first one is + and others are numbers, then trim + and return remaining 10 digits.
- If the phone number contains + symbol more than one, consider it as a bad number.
- If the phone number contains + symbol not as the first character, consider it as a bad number.

Ignore spaces between digits.

Input	Output
“455678”	“Bad number”
“339 5656888”	“3395656888”
“+0008989562”	“0008989562”
“45231489562”	“Bad number”
“123+2356897452”	“Bad number”

10. Write a function to convert a long phrase to its acronym.

Input	Output
‘Prisoner of War’	‘POW’
‘Have a good night’	‘HAGN’

11. Write a function, which receives an array as an argument which elements arrays of numbers. Find product of biggest negative number of each array. If there is not any negative number, ignore that array. Check that items of the given array are arrays.

Input	Output
[[2, -9, -3, 0], [1, 2], [-4 , -11, 0]]	12
[[3, 4], [11, 0], [5, 6, 7, 8]]	‘No negatives’
[1, 2, 3]	‘Invalid Argument’

12. 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’] }

13. 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.

14. Given two objects. Write a function that performs shallow compare.

Input	Output

let a = { a: ‘1’ };
let b = { a: ‘1’};
shallowCompare(a, b)
	true

let a = { a: ‘1’ };
let b = { a: ‘1’, b: ‘2’ };
shallowCompare(a, b)
	false

let a = { };
let b = { };
shallowCompare(a, b)
	true

15. Write a function which calculator average age of user (your function must use reduce method).

```
const users = [
  {
    name: "John Doe",
    age: 30,
    isAdmin: false,
  },
  {
    name: "Jane Smith",
    age: 25,
    isAdmin: true,
  },
  {
    name: "Bob Johnson",
    age: 40,
    isAdmin: false,
  },
];

getAverageAge(users);
```

16. Given an array of cars with properties like make, model, and year, print out the car's details in a readable format for given care make.

```
const cars = [
  {
    make: "Toyota",
    model: "Corolla",
    year: 2022,
  },
  {
    make: "Honda",
    model: "Civic",
    year: 2023,
  },
  {
    make: "Ford",
    model: "Mustang",
    year: 2021,
  },
  {
    make: "Toyota",
    model: "Camry",
    year: 2023,
  },
  {
    make: "Chevrolet",
    model: "Silverado",
    year: 2022,
  },
  {
    make: "Toyota",
    model: "Rav4",
    year: 2023,
  },
];

//input: "Toyota"

// output: "Make: Toyota, Model: Corolla, Year: 2022" , "Make: Toyota,
Model: Camry, Year: 2023", "Make: Toyota, Model: Rav4, Year: 2023"
```

17. Given two objects representing rectangles with properties width and height, write a function to compare if they have the same area.

```
const rectangle1 = {
  width: 5,
  height: 10,
};

const rectangle2 = {
  width: 10,
  height: 5,
};

const rectangle3 = {
  width: 6,
  height: 8,
};

areRectanglesSameArea(rectangle1, rectangle2)); // Output: true
areRectanglesSameArea(rectangle1, rectangle3)); // Output: false
```

18. Given an object representing a bank account with properties like accountNumber, balance, and accountHolder. Write functions to deposit and withdraw money from the account.

```
const bankAccount = {
  accountNumber: "123456789",
  balance: 1000,
  accountHolder: "John Doe",
};

updateBalance(500, "deposit"); // Output: Deposited $500. New balance: $1500
updateBalance(200, "withdraw"); // Output: Withdrew $200. New balance: $1300
updateBalance(-100, "deposit"); // Output: Invalid deposit amount. Please deposit a positive amount.
updateBalance(5000, "withdraw"); // Output: Insufficient balance or invalid withdrawal amount.
```