

4. 알고리즘 연습 문제

#1. 짝수와 홀수

evenOrOdd 함수는 정수 num을 매개변수로 받는다. num은 1이상의 정수이며, num이 음수인 경우는 없다. num이 짝수일 경우 'Even'을 반환하고 홀수인 경우 'Odd'를 반환하도록 evenOrOdd 함수를 완성하라.

단, if문을 사용한 답과 삼항 조건 연산자를 사용한 답 두가지를 제시하여야 한다.

JAVASCRIPT

```
// if문
function evenOrOdd(num) {

}

console.log(evenOrOdd(2)); // Even
console.log(evenOrOdd(3)); // Odd
console.log(evenOrOdd(1000)); // Even

// 3항 연산자
```

```
function evenOrOdd(num) {  
  
}  
  
console.log(evenOrOdd(2)); // Even  
console.log(evenOrOdd(3)); // Odd  
console.log(evenOrOdd(1000)); // Even
```

2. 1 ~ 10,000의 숫자 중 8이 등장하는 횟수 구하기 (Google)

1부터 10,000까지 8이라는 숫자가 총 몇번 나오는가? 이를 구하는 함수를 완성하라.

단, 8이 포함되어 있는 숫자의 갯수를 카운팅 하는 것이 아니라 8이라는 숫자를 모두 카운팅 해야 한다. 예를 들어 8808은 3, 8888은 4로 카운팅 해야 한다.

(hint) 문자열 중 n번째에 있는 문자 : str.charAt(n) or str[n]

JAVASCRIPT

```
function getCount8 () {  
  
}  
  
console.log(getCount8()); // 4000
```

3. 문자열 다루기

alphaString46 함수는 문자열 s를 매개변수로 입력받는다. s의 길이가 4 ~ 6이고, 숫자로만 구성되어 있는지 확인하는 alphaString46 함수를 완성하라.

예를 들어 s가 'a234'이면 false를 리턴하고 '1234'라면 true를 리턴한다.

JAVASCRIPT

```
function alphaString46(s) {  
  
}  
  
console.log(alphaString46('1234')); // true  
console.log(alphaString46('9014')); // true  
console.log(alphaString46('723')); // false  
console.log(alphaString46('a234')); // false  
console.log(alphaString46('')); // false  
console.log(alphaString46()); // false
```

4. 문자열 내 p와 y의 개수

numPY함수는 대문자와 소문자가 섞여있는 문자열 s를 인수로 전달받는다. s에 존재하는 'p'의 개수와 'y'의 갯수를 비교해 같으면 true, 다르면 false를 리턴하도록 함수를 완성하라. 대소문자를 구별하지 않으며 'p', 'y' 모두 하나도 없는 경우는 항상 true를 리턴한다.

예를 들어 s가 'pPoooyY'면 true를 리턴하고 'Pyy'라면 false를 리턴한다.

JAVASCRIPT

```
function numPY(s) {  
  
}
```

```
console.log(numPY('pPooyY')); // true
console.log(numPY('PyY'));    // false
console.log(numPY('ab'));     // true
console.log(numPY(''));       // true
console.log(numPY());         // true
```

5. 이상한 문자 만들기

toWeirdCase함수는 문자열을 인수로 전달받는다. 문자열 s에 각 단어의 짝수번째 인덱스 문자는 대문자로, 홀수번째 인덱스 문자는 소문자로 바꾼 문자열을 리턴하도록 함수를 완성하라.

예를 들어 s가 'hello world'라면 첫번째 단어는 'HeLlO', 두번째 단어는 'WoRlD'로 바뀌어 'HeLlO WoRlD'를 리턴한다.

주의) 문자열 전체의 짝/홀수 인덱스가 아니라 단어(공백을 기준)별로 짝/홀수 인덱스를 판단한다.

JAVASCRIPT

```
function toWeirdCase(s) {

}

console.log(toWeirdCase('hello world'));    // 'HeLlO WoRlD'
console.log(toWeirdCase('my name is lee')); // 'My NaMe Is LeE'
```

6. 핸드폰번호 가리기

핸드폰 요금 고지서에 표시할 전화번호는 개인정보 보호를 위해 맨 뒷자리 4자리를 제외한 나머지를 * 으로 바꿔야 한다. 전화번호를 나타내는 문자열 str을 입력받는 hideNumbers 함수를 완성하라 예를들어 s가 '01033334444'면 *****4444 를 리턴하고, '027778888'인 경우는 *****8888 을 리턴한다.

JAVASCRIPT

```
function hideNumbers(str) {  
  
}  
  
console.log(hideNumbers('01033334444')); // *****4444  
console.log(hideNumbers('027778888')); // *****8888
```

7. 문자열을 숫자로 바꾸기

strToInt 메소드는 문자열을 인수로 전달받는다. 전달받은 문자열 인수를 숫자로 변환한 결과를 반환하도록 strToInt를 작성하라.

예를 들어 str이 '1234'이면 1234를 반환하고, '-1234'이면 -1234를 반환한다. str은 부호(+,-)와 숫자로만 구성되어 있고, 잘못된 값이 입력되는 경우는 없다.

JAVASCRIPT

```
function strToInt(str) {  
  
}  
  
console.log(strToInt('1234')); // 1234  
console.log(strToInt('-1234')); // -1234
```

8. 수박수박수박수박수박수?

waterMelon 함수는 정수를 인수로 전달받는다. 길이가 n이고, 수박수박수...와 같은 패턴을 유지하는 문자열을 리턴하도록 함수를 완성하라.

예를 들어 n이 4이면 '수박수박'을 리턴하고 3이라면 '수박수'를 리턴한다.

JAVASCRIPT

```
function waterMelon(n) {  
  
}  
  
console.log('n이 3인 경우: ' + waterMelon(3));  
console.log('n이 4인 경우: ' + waterMelon(4));
```

9. 정수제곱근 판별하기

nextSqaure 함수는 정수를 인수로 전달받는다. n이 임의의 정수 x의 제곱이라면 x+1의 제곱을 리턴하고, n이 임의의 정수 x의 제곱이 아니라면 'no'를 리턴하는 함수를 작성하라.

예를 들어 n이 121이라면 이는 정수 11의 제곱이므로 (11+1)의 제곱인 144를 리턴하고, 3이라면 'no'를 리턴한다.

JAVASCRIPT

```
function nextSqaure(n){  
  
}
```

```

console.log(nextSqaure()); // no
console.log(nextSqaure(0)); // 1
console.log(nextSqaure(1)); // 4
console.log(nextSqaure(2)); // no
console.log(nextSqaure(3)); // no
console.log(nextSqaure(121)); // 144
console.log(nextSqaure(165)); // no
console.log(nextSqaure(400)); // 441

```

10. 배열의 최대/최소값 구하기

배열의 요소 중 최대값/최소값을 반환하는 함수를 완성하라.

JAVASCRIPT

```

function getMaxValueFromArray(array) {

}

console.log(getMaxValueFromArray([3, 6, -2, -5, 7, 3])); // 7

function getMinValueFromArray(array) {

}

console.log(getMinValueFromArray([3, 6, -2, -5, 7, 3])); // -5

```

11. Check Palindrom

palindrome(팰린드롬/회문)은 왼쪽에서 오른쪽으로 읽은 다음, 오른쪽부터 왼쪽으로 다시 읽

어도 똑같은 형태와 의미를 유지하는 문장이나 단어를 지칭한다. 인자로 전달한 문자열이 palindrome인지 검사하여 Boolean값을 반환하는 함수를 완성하라. 단, 반드시 1자 이상의 문자열을 인자로 전달한다.

JAVASCRIPT

```
function checkPalindrom(str) {  
  
}  
  
console.log(checkPalindrom('dad')); // true  
console.log(checkPalindrom('mom')); // true  
console.log(checkPalindrom('palindrom')); // false  
console.log(checkPalindrom('s')); // true
```

#12. 중복 요소 제거

인수로 전달된 배열의 요소 중에서 중복된 요소를 제외하고 유니크한 요소만을 반환하는 함수를 작성하라.

for 문은 사용하지 않도록 하자.

JAVASCRIPT

```
function uniq(array) {  
  
}  
  
console.log(uniq([2, 1, 2, 3, 4, 3, 4])); // [ 2, 1, 3, 4 ]
```


#13. 중복없는 배열

길이가 n 인 배열에 1부터 n 까지 숫자가 중복 없이 한 번씩 들어 있는지를 확인하려고 한다. 1부터 n 까지 숫자가 중복 없이 한 번씩 들어 있는 경우 `true`를, 아닌 경우 `false`를 반환하도록 함수 `isNotOverlapArray`을 완성하라. 단, 배열의 요소는 정수이다.

예를 들어 주어진 배열이 `[4, 1, 3, 2]`이라면 `true`, `[4, 1, 3]` 또는 `[1, 3]`이라면 `false`를 반환한다.

JAVASCRIPT

```
function isNotOverlapArray(array) {  
  
}  
  
console.log(isNotOverlapArray([4, 1, 3, 2])); // true  
console.log(isNotOverlapArray([4, 1, 3]));    // false
```

#14. 중복된 요소

인수로 전달된 배열의 요소 중에서 중복된 요소만으로 구성된 배열을 반환하는 함수를 작성하라.

`for` 문은 사용하지 않도록 하자.

JAVASCRIPT

```
function findDuplicated(array) {  
  
}  
  
console.log(findDuplicated([1, 2, 3, 4, 1, 2, 3])); // [ 1, 2, 3 ]
```

#15. 약수의 합

어떤 수를 입력받아 그 수의 약수를 모두 더한 수를 구하는 `sumDivisor` 함수를 완성하라. 예를 들어 12가 입력된다면 12의 약수는 [1, 2, 3, 4, 6, 12]가 되고, 총 합은 28이 되므로 28을 반환한다.

약수(約數, divisor)는 어떤 수를 나누었을 때 나머지가 0인 수를 말하며, 배수 관계와 서로 반대되는 개념이다

JAVASCRIPT

```
function sumDivisor(num) {  
  
}  
  
console.log(sumDivisor(12)); // 28
```

#16. 소수 찾기

`numberOfPrime` 메소드는 정수 `n`을 매개변수로 입력받는다. 1부터 입력받은 숫자 `n` 사이에 있는 소수의 개수를 반환하도록 `numberOfPrime` 함수를 완성하라. 예를 들어 10을 입력받았다면, 1부터 10 사이의 소수는 [2,3,5,7] 4개가 존재하므로 4를 반환한다.

소수(素數, prime number)는 2, 3, 5, 7, 11, 13, 17...과 같이 1과 자신 이외의 어떤 수로도 나눌 수 없는 1보다 큰 양의 정수이다.

JAVASCRIPT

```
function numberOfPrime(n) {  
  
}  
  
console.log(numberOfPrime(10)); // 4
```

17. 피보나치 수

피보나치 수는 0과 1로 시작하며, 다음 피보나치 수는 바로 앞의 두 피보나치 수의 합이 된다. 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946...

2 이상의 n 이 입력되었을 때, n 번째 피보나치 수를 반환하는 fibonacci 함수를 작성하라. 예를 들어 $n = 3$ 이라면 2를 반환한다.

JAVASCRIPT

```
function fibonacci(n) {  
  
}  
  
console.log(fibonacci(2)); // 1  
console.log(fibonacci(3)); // 2  
console.log(fibonacci(4)); // 3  
console.log(fibonacci(5)); // 5  
console.log(fibonacci(6)); // 8
```

18. 각 자릿수의 합 구하기

정수 n 이 주어지면, n 의 각 자릿수의 합을 구해서 반환하는 `digitSum` 함수를 완성하라. 예를 들어 $n = 123$ 이면 $1 + 2 + 3 = 6$ 을 반환한다. 단, n 은 100,000,000 이하의 정수로 한다.

JAVASCRIPT

```
function digitSum(n) {  
  
}  
  
console.log(digitSum(123)); // 6  
console.log(digitSum(987)); // 24  
console.log(digitSum(100000001)); // false
```

19. 하샤드 수

하샤드 수는 그 수의 각 자릿수 숫자의 합으로 그 수가 나누어지는 양의 정수를 말한다.

양의 정수 x 가 하샤드 수이라면 x 의 자릿수의 합으로 x 가 나누어져야 한다. 예를들어 18의 자릿수 합은 $1+8=9$ 이고, 18은 9로 나누어 떨어지므로 18은 하샤드 수이다.

10, 12, 18, 20, 21, 24, 27, 30, 36, 40, 42, 45, 48, 50, 54, 60, 63, 70, 72, 80, 81, 84, 90,
100, 102, 108, 110, 111, 112, 114, 117, 120, 126, 132, 133, 135, 140, 144, 150, 152, 153, 156,
162, 171, 180, 190, 192, 195, 198, 200

Harshad함수는 양의 정수 n 을 매개변수로 입력받는다. 이 n 이 하샤드수인지 아닌지 판단하는 함수를 완성하라.

예를 들어 n이 10, 12, 18이면 True를 리턴 11, 13이면 False를 리턴한다.

JAVASCRIPT

```
function isHarshad(n){  
  
}  
  
console.log(isHarshad(10)); // true  
console.log(isHarshad(12)); // true  
console.log(isHarshad(18)); // true  
console.log(isHarshad(11)); // false  
console.log(isHarshad(13)); // false
```

20. 배열의 첫 요소와 마지막 요소로 배열 만들기

배열의 첫 요소와 마지막 요소를 나타내는 정수를 인자로 받아 정수의 배열을 반환하는 함수를 완성하라. 예를 들어 인수가 [10, 15]인 경우, [10, 11, 12, 13, 14, 15]를 반환한다.

JAVASCRIPT

```
function generateRange(from, to) {  
  const res = [];  
  
  return res;  
}  
  
console.log(generateRange(10, 15)); // [ 10, 11, 12, 13, 14, 15 ]
```

21. 두 정수 사이의 합

adder 함수는 정수 x, y를 인수로 전달받는다. 두 수와 두 수 사이에 있는 모든 정수를 더해서 리턴하도록 함수를 완성하라.

x와 y가 같은 경우는 둘 중 아무 수나 리턴한다. x, y는 음수나 0, 양수일 수 있으며 둘의 대소 관계도 정해져 있지 않다.

예를 들어 x가 3, y가 5이면 12를 리턴한다.

JAVASCRIPT

```
function adder(x, y){  
  
}  
  
console.log(adder(3, 5)); // 12
```

22. 배열의 인접한 요소곱 중 가장 큰 값 구하기

인수로 주어진 정수의 배열에서 인접한 요소의 곱이 가장 큰 값을 반환하는 함수를 완성하라.

예를 들어 인수가 [3, 6, -2, -5, 7, 3]인 경우, 21을 반환한다.

JAVASCRIPT

```
function adjacentElementsProduct(arr) {  
  
}  
  
console.log(adjacentElementsProduct([3, 6, -2, -5, 7, 3])); // 21
```

23. 배열에서 특정 값만을 구하기

인수로 주어진 배열 arr에서 짝수이고 3보다 큰 수만을 구하여 이를 배열로 반환하는 함수를 작성하라

JAVASCRIPT

```
function getArray(arr) {  
  
}  
  
console.log(getArray([1, 2, 3, 4, 5, 6])); // [ 4, 6 ]
```

24. 평균구하기

인수로 주어진 배열의 평균을 구하는 함수를 완성하라.

JAVASCRIPT

```
function average(array){  
  
}  
  
console.log(average([5, 3, 4])); // 4
```

25. 최단 거리 1차원 점의 쌍 구하기 (DAUM)

1차원의 점들이 주어졌을 때, 그 중 가장 거리가 짧은 것(들)의 쌍을 배열로 반환하는 함수를 작성하라. (단 점들의 배열은 모두 정렬되어있다고 가정한다.) 예를들어 [1, 3, 4, 8, 13, 17, 20, 23, 24]이 주어졌다면, 결과값은 [[3, 4], [23, 24]]가 될 것이다.

JAVASCRIPT

```
function findMinDistance(array){  
  
}  
  
// 1차원 점의 배열  
var array = [1, 3, 4, 8, 13, 17, 20, 23, 24];  
console.log(findMinDistance(array)); // [[3, 4], [23, 24]]
```

26. 특별한 정렬

n개의 정수를 가진 배열이 있다. 이 배열은 양의 정수와 음의 정수를 모두 가지고 있다. 이 배열을 좀 특별한 방법으로 정렬해야 한다. 음의 정수는 앞쪽에 내림차순으로, 양의 정수는 뒷쪽에 있어야 한다. 단, 인수로 주어진 원본 배열은 변경되지 않아야 한다.

예를 들어, [-1, 1, 3, -2, 2, 0]이 주어졌을 때, [-1, -2, 0, 1, 2, 3]를 반환한다.

JAVASCRIPT

```
function specialSort(array) {  
  
}
```



```
const testArray = [-1, 1, 3, -2, 2, 0];

console.log(testArray); // [ -1, 1, 3, -2, 2, 0 ]
console.log(specialSort(testArray)); // [ -1, -2, 0, 1, 2, 3 ]
```

27. 요일 구하기

2016년 1월 1일은 금요일이다. 2016년 a월 b일은 무슨 요일일까? 두 수 a, b를 입력받아 a월 b일이 무슨 요일인지 출력하는 `getDayName` 함수를 완성하라.

요일의 이름은 일요일부터 토요일까지 각각 SUN, MON, TUE, WED, THU, FRI, SAT를 출력한다. 예를 들어 a=5, b=24가 입력된다면 5월 24일은 화요일이므로 TUE를 반환한다.

JAVASCRIPT

```
function getDayName(a, b){

}

console.log(getDayName(5, 24)); // TUE
```