# Задачи на логику

## 1. У вас есть восемь монет и одни весы с двумя чашами. Одна монета немного тяжелее остальных. Как за два взвешивания найти тяжелую монету?

### Делим монеты на 3 кучи , 3-3-2 , дальше взвешиваем 3-3 , если одна тяжелее взвешиваем 2 монеты из трёх , если одна тяжелее , тогда мы нашли нашу монету , иначе 3-я монета оставшая не взвешенной и есть нужная. В случае если кучи монет 3-3 одинаковые по весу , тогда проверяем кучу из двух монет , делим ее по одной монете и взвешиваем

## 2.

## 3.Ученые разработали новый материал неизвестной прочности. Они знают, что материал разбивается при падении с высоты от 1 метра до 5 000 метров. Но не знают, с какой именно высоты. Чтобы определить прочность, ученые поднимают предмет на некоторую высоту и сбрасывают его оттуда. Их задача — определить, начиная с какой именно высоты предмет начнет разбиваться.

### берем первый предмет начинаем кидать с высоты 1 метр  , далее увеличиваем высоту до 50 метров , затем увеличиваем на 50 каждый раз и так до тех пор пока предмет не разобьется , после того как предмет разбился берем предыдущий шаг и увеличивая его на 1 ищем нужную высоту.

### в лучшем случаее мы найдем высоту с 1 попытки , в худшем поиск займет 151 попытку

## 4. В университетскую столовую пришли трое студентов купить пирожки. Один хочет купить треть всех имеющихся в столовой пирожков и еще 2 пирожка. Другой хочет купить четверть от всех имеющихся пирожков и ещё 3 пирожка. А третий хочет купить пятую часть всех пирожков и ещё 8 пирожков. Если все трое купят столько пирожков, сколько планируют, то в столовой как раз закончатся все пирожки. Сколько пирожков есть в столовой?

### Получаем уровнение 1/3x + 2 + 1/4x + 3 + 1/5x + 8  = x

### Далее приводим к общему множителю 20x/60 + 15x/60 + 12x/60 + 13 = x

### 47x/60 + 13  = x , переносим 47x/60 вправо со знаком минус

### 13 = x - 47x/60

### 13 = 13x/60

### 13 \* 60 = 13x  ==>   780 = 13x   ===>  60 = x . Получаем 60 пирожков

# JavaScript Functions

## 1. Функция  которая принимает целое число в качестве аргумента и возвращает строку, содержащую это число и слово "компьютер" в нужном склонении по падежам в зависимости от числа

```javascript

const getComputersCountString = (count) => {

    let decimalPlace = count >= 10 ? count % 10 : Math.abs(count)

    if(decimalPlace === 0 || decimalPlace >= 5)

        return `${count} компьютеров`

    else if(decimalPlace === 1)

        return `${count} компьютер`

    else if(decimalPlace > 1 && decimalPlace < 5)

        return `${count} компьютера`

    return -1

}

```

## 2. Функция которая на вход получает массив положительных целых чисел произвольной длины. Например [42, 12, 18]. На выходе возвращает массив чисел, которые являются общими делителями для всех указанных числе

```

const getArrayDivisors = (arr)=>{

    if(!Array.isArray(arr) || arr.length == 0 ) return []

    const minValueInArray = Math.min(...arr)

    let divisors = []

    for(let i = 2 ; i < minValueInArray ; i++){

        if(minValueInArray % i == 0) divisors.push(i);

    }

    for(let i = 0; i < arr.length ; i++){

        divisors = divisors.filter((divisor) => arr[i] % divisor == 0)

    }

    return divisors

}

```

## 3. Функция которая возвращает массив простых чисел в диапазоне (2 числа - минимальное и максимальное) заданных чисел

```

const isPrime = (num) => {

    if (num < 2) return false

    if (num >=2 && num < 4) return true;

    for(let i = 2; i < num; i+=1)

        if(num % i == 0)return false

    return true

}

const getPrimeNumbers = (start , end)=>{

    let result = []

    for(let i = start ; i < end ; i++){

        isPrime(i) && result.push(i)

    }

    return result

}

```

## 4. Функция для вывода таблицы умножения

```

const printMultiplicationTable = (value) =>{

    if (!value && value < 2) {

        console.log('invalid value')

        return

    }

    let result = "\t"

    for(let i = 1; i <= value ; i ++)

        result += i + "\t"

    for(let i = 1; i <= value; i++){

        result +='\n'+ i + "\t"

        for(let j = 1; j <= value; j++)

            result += (i \* j) + "\t"

    }

    console.log(result)

}

```