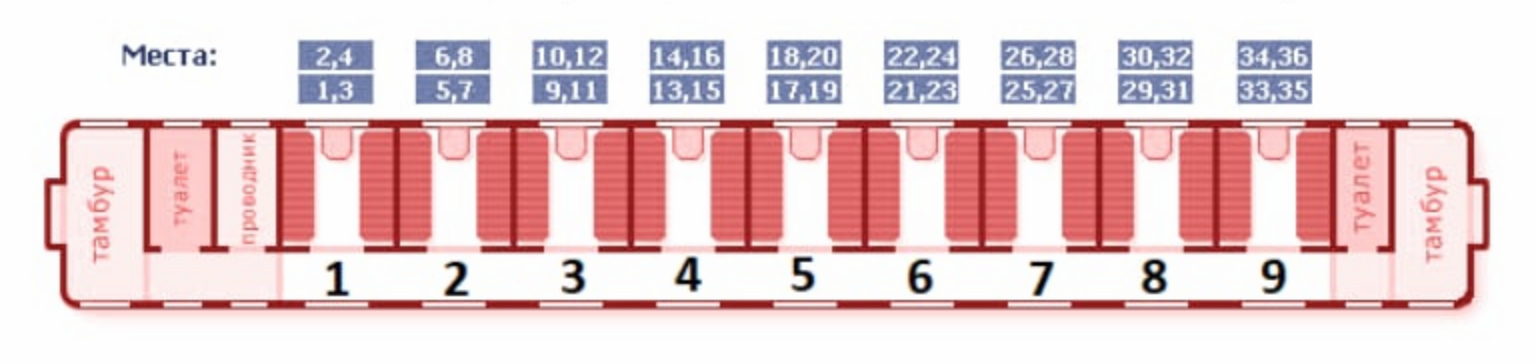
Задачи:

1) Создайте функцию, которая будет вычислять объем и площадь **полной поверхности** куба (базовая математика, иногда используется в создании анимаций). Эта функция принимает в себя целое **число** со значением длины ребра куба. Ответ выведите в формате **строки**, который изображен в примерах.

Если в функцию попал неправильный аргумент или вычислить значения невозможно - вернуть строку "При вычислении произошла ошибка"

**НЕ ИСПОЛЬЗУЙТЕ ОПЕРАТОР СТЕПЕНИ** \*\*

Примеры:  
calculateVolumeAndArea(5)  => 'Объем куба: 125, площадь всей поверхности: 150'  
calculateVolumeAndArea(15)  => 'Объем куба: 3375, площадь всей поверхности: 1350'  
calculateVolumeAndArea(15.5)  => 'При вычислении произошла ошибка'  
calculateVolumeAndArea('15')  => 'При вычислении произошла ошибка'  
calculateVolumeAndArea(-15)  => 'При вычислении произошла ошибка'

2) Напишите функцию, которая будет определять номер купе по переданному ей номеру места. Наглядно:Функция принимает **только** целое число от 1 до 36.  
Если переданный аргумент не число, отрицательное или дробное - **возвращается** сообщение:  
"Ошибка. Проверьте правильность введенного номера места"  
Если число 0 или больше 36, то сообщение: "Таких мест в вагоне не существует"  
Пример:  
getCoupeNumber(33)  => 9  
getCoupeNumber(7)  => 2  
getCoupeNumber(300)  => "Таких мест в вагоне не существует"  
getCoupeNumber(0)  => "Таких мест в вагоне не существует"  
getCoupeNumber(7.7)  => "Ошибка. Проверьте правильность введенного номера места"  
getCoupeNumber(-10)  => "Ошибка. Проверьте правильность введенного номера места"  
getCoupeNumber('Hello')  => "Ошибка. Проверьте правильность введенного номера места"  
Такая функция вполне реальна и может использоваться для формирования билетов, в том числе и визуально на сайтах. Конечно, там будет куда больше условий, но смысл остается таким же.

3) Создайте функцию, которая принимает в себя целое число минут и **возвращает** время в нужном формате **строки**. (Смотри пример). Обратите внимание на окончание слова "час" - оно меняется в зависимости от цифры. Если вместо аргумента приходит не число, дробное или отрицательное число - функция возвращает строку "Ошибка, проверьте данные"

**Внимание!** Давайте пока ограничимся **максимум** 600ю минутами (10 часов). Так как проверки на большие числа будут раздувать код (33 часа, 31 час, 11 часов и тд). Этого будет достаточно и код будет проверять именно этот промежуток (1 - 10 часов). Но вы можете реализовать и полный скрипт, он тоже должен проходить тесты.

Пример:  
getTimeFromMinutes(150) => "Это 2 часа и 30 минут"  
getTimeFromMinutes(50) => "Это 0 часов и 50 минут"  
getTimeFromMinutes(0) => "Это 0 часов и 0 минут"  
getTimeFromMinutes(-150) => "Ошибка, проверьте данные"

4) Напишите функцию, которая принимает в себя 4 числа и возвращает самое большее из них. Если один из аргументов не является числом или их меньше 4 - **возвращается** 0. Дробные числа разрешены.  
Пример:  
findMaxNumber(1, 5, 6.6, 11); =>  11  
findMaxNumber(1, 5, '6', '10');  =>  0

**(\*\*) Задача с собеседований на числа Фибоначчи**

Это одна из классических задач в программировании на самых разных языках. Скорее всего вы слышали про [числа Фибоначчи](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8), где первые два числа равны 0 и 1, а каждое последующее число равно сумме двух предыдущих чисел.

Подсказку (но не решение этой задачи) можно найти [тут](https://learn.javascript.ru/task/fibonacci-numbers). Но постарайтесь не открывать 🙂

Задача:

Создайте функцию, которая будет принимать в себя один аргумент-целое положительное число. Она должна **возвращать** строку, в которой будут через пробел выведены числа Фибоначчи. Причем, их количество должно быть равно переданному аргументу.

Если переданный аргумент не число - вернуть пустую строку. Решать **без применения рекурсии**.

Пример:

fib(4) => ''0 1 1 2"

fib(7) => ''0 1 1 2 3 5 8"

fib('7') => ''"

fib(1) => "0"

fib(0) => ''"

Задача **непростая** для новичков, так что с первого раза может не получится.