Напишите на языке Haskell определения следующих функций:

1. Написать функцию bitOnes :: Integer -> Int, которая выдает количество единиц в битовом представлении заданного натурального числа. Например, число 36 имеет битовое представление 100100, поэтому bitOnes 36 => 2. Для решения пользуйтесь арифметическими операциями с целыми числами.
2. Написать функцию ch :: Double -> Int -> Double, которая должна вычислять значение гиперболического косинуса в заданной точке путем суммирования некоторого числа первых членов ряда в разложении этой функции в ряд Тейлора. Количество суммируемых членов задается вторым аргументом функции.  
   *Указание*: для проверки правильности результатов можно воспользоваться стандартной функцией cosh.
3. Написать функцию segmEquals :: Eq a => [a] -> [a], которая в списке значений находит отрезок максимальной длины подряд стоящих равных значений. Например, в списке [1,3,3,6,7,3] таким отрезком будет [3,3], а в списке [5,5,5,1,1,1] результатом может быть любой из двух списков [5,5,5] или [1,1,1]. Для пустого списка результатом будет также пустой список..  
   Замечание. Eq a – ограничение на тип, которое позволяет для сравнения значений использовать операции (==) и (/=)

Решения следует передавать в Moodle в виде файла (или файлов) с исходным текстом на языке Haskell.