

1. [1 балл] Предикат - это функция, которая проверяет, выполняется ли определенное условие для переданного ей значения. Напишите функцию, которая принимает два предиката и возвращает новый предикат, который реализует их логическое И. Этот новый предикат должен возвращать `True` только в том случае, если для переданного значения оба исходных предиката возвращают `True`. В противном случае он должен возвращать `False`.

```
combinePredicates :: (Int -> Bool) -> (Int -> Bool) -> (Int -> Bool)
combinePredicates p1 p2 = undefined
```

Например:

```
ghci> p = combinePredicates even (>10)
ghci> p 11
False
ghci> p 12
True
ghci> p 2
False
```

2. [1 балл] Реализуйте функцию, которая принимает функцию f , её аргумент x , и малое значение h , и возвращает приближённое значение производной в точке x с использованием конечных разностей: [Численное дифференцирование — Википедия](#)

```
differentiate :: (Double -> Double) -> Double -> Double -> Double
differentiate f x h = undefined
```

3. [3 балла] Напишите функцию `solver` для нахождения корня уравнения вида $f(x) = 0$ методом [деления пополам](#) на отрезке $[a; b]$ с заданной точностью ε (т.е. функция должна находить x_0 : $|f(x_0)| < \varepsilon$).

Функция `solver` должна принимать следующие аргументы:

1. Непрерывную функцию, корень которой нужно найти
2. Точность ε
3. Границы отрезка (не забудьте убедиться, что на границах функция разного знака)