**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе**

По дисциплине «СиАОД»

Выполнил: студент группы БВТ1801

Задоркин Максим Александрович

Руководитель:

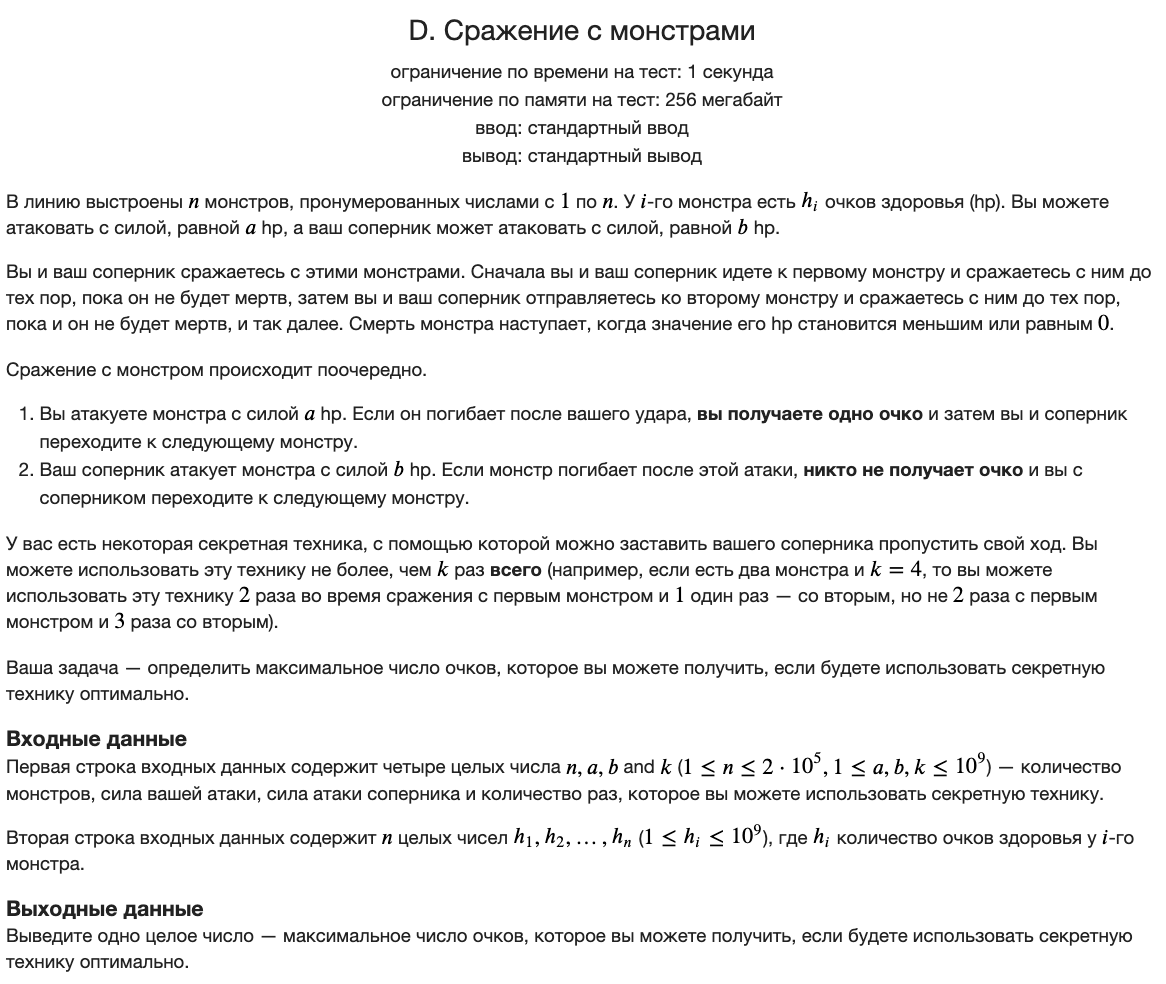
Кутейников Иван Алексеевич

Москва 2020

**Задача:**Решить несколько задач (не менее 2) уровнем не ниже 1300-1400 с сайта codeforces.com

**Ход работы:**

Код так же можно посмотреть на моем github: https://github.com/MaximZar/SaDPA/tree/master/codeForces

**Задача №1**  
Условие задачи: <https://codeforces.com/problemset/problem/1296/D>  
Сложность: 1500  


Код решения и пояснение к решению:

import **java.util.ArrayList**;

import **java.util.Collections**;

import **java.util.Scanner**;

*// https://codeforces.com/problemset/problem/1296/D*

**public** **class** Task1 {

**public** **static** **void** main(**String**[] args) {

**Scanner** input = **new** Scanner(System.in);

**int** n = input.nextInt(); *// количество монстров*

**double** a = input.nextInt(); *// моя сила удара*

**double** b = input.nextInt(); *// сила удара соперника*

**int** k = input.nextInt(); *// количество возможных использований секретной техники*

**double** sumDamage = a + b; *// общая сила удара*

**ArrayList**<**Integer**> monsters = **new** **ArrayList**<**Integer**>(); *// массив для монстров, которых не победили(дальше поподробнее)*

**int** countWin = 0; *// количество возможных выигрышных партий, очков*

**for** (**int** i = 1; i <= n; i += 1) {

**double** monster = input.nextInt(); *// очередной монстр*

**double** remainingDamage = monster % sumDamage; *// остаток урона после серии ударов*

*// remainingDamage = по сути ищем сколько останется хп перед итоговым боем*

*// или победил ли его наш соперник. Далее станет яснее*

*// последний удар сделали не мы, поэтому этот случай можно обработать*

*// для этого откатимся на 1 пару ходов. А потом будем использовать технику.*

**if** (remainingDamage == 0) {

*// количество проведенных секретных техник понадобится*

*// (на основе оставшегося hp у монстра при откате на 1 пару ходов):*

**int** amountOfSecretTechnology = (**int**) Math.ceil(b / a);

monsters.add(amountOfSecretTechnology);

**continue**;

}

**if** (remainingDamage <= a) { *// проверяем сможем ли именно МЫ его "добить" c 1 удара*

countWin += 1; *// прибавляем очко за победу*

**continue**;

}

*/\* в данном случае количество необходимых проведенных техник вычисляется несколько иначе*

*\* учитываем то, что монстр не был добит (remainingDamage). А значит нужно теперь пользоваться техниками*

*\* чтобы добить первее без отката на шаг назад*

*\*/*

**int** amountOfSecretTechnology = (**int**) Math.ceil((remainingDamage - a) / a);

monsters.add(amountOfSecretTechnology);

}

Collections.sort(monsters); *// сортируем чтобы выбрать первых двух с самым малым количеством*

*// использования секретных техник для выигрыша*

**int** index = 0; *// просто счетчик индексов*

**int** sizeOfMonsters = monsters.size();

**while** (k > 0) {

**if** (index >= sizeOfMonsters) **break**;

*// берем слабенького монстра*

**int** monster = monsters.get(index);

*/\* если запас техник остался то минусуем и добавляем очко выигрыша*

*\* если нет то прекращаем,*

*\* так как дальше монстры с еще большим количеством*

*\* необходимых техник для победы и смысла смотреть нет*

*\*/*

**if** (k - monster >= 0) {

k -= monster;

countWin += 1;

}

**else** **break**;

index += 1;

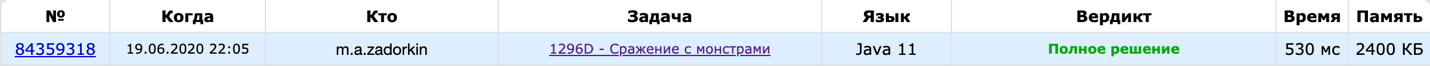
}

System.out.println(countWin);

}

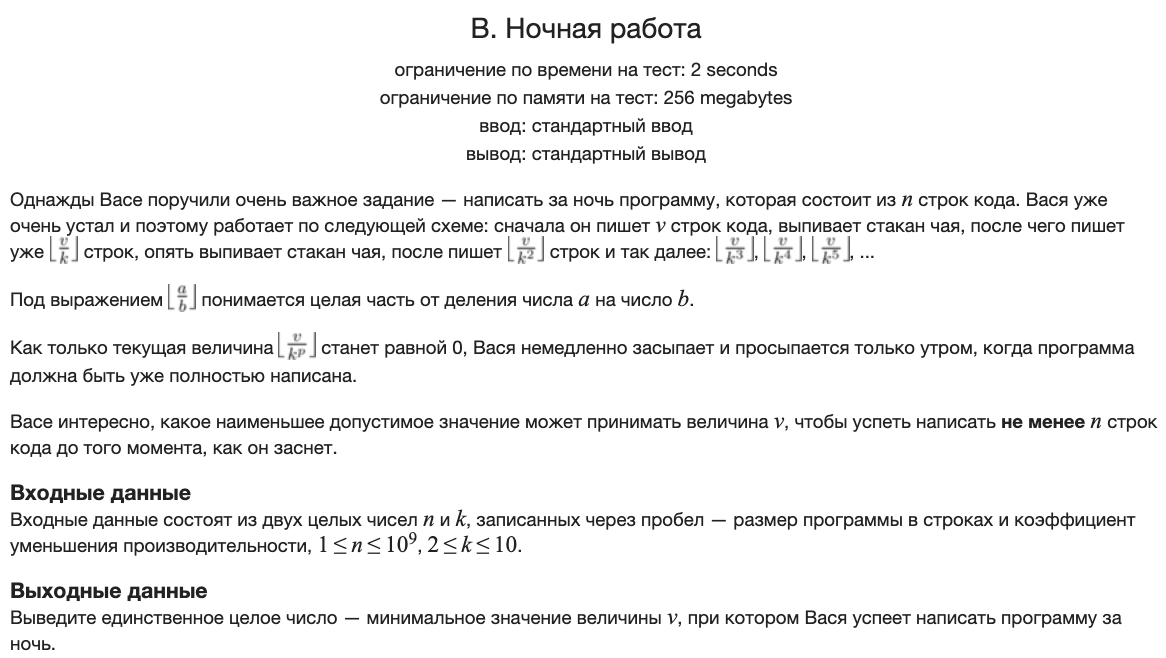
}

Результат тестирования:



**Задача 2:**

Условие задачи: <https://codeforces.com/problemset/problem/165/B>  
Сложность: 1500



Код задачи и пояснение к решению:

import **java.util.Scanner**;

*// https://codeforces.com/problemset/problem/165/B*

**public** **class** Task2 {

**public** **static** **void** main(**String**[] args) {

**Scanner** input = **new** Scanner(System.in);

**int** n = input.nextInt(); *// строк нужно написать*

**int** k = input.nextInt(); *// коэффициент уменьшения*

*/\* число v будет располагаться в пределах 1 .. n*

*\* а значит можем воспользоваться бинарным поиском*

*\* чтобы эффективнее найти нужную цифру*

*\* для реализации бинарного поиска создадим leftPoint и rightPoint*

*\* чтобы помечать наши отрезки*

*\*/*

**int** leftPoint = 1;

**int** rightPoint = n;

**int** v = n;

**while** (leftPoint < rightPoint) {

**int** pivot = leftPoint + (rightPoint - leftPoint) / 2; *// v на этой итерации,*

*// середина нашего отрезка текущего*

**int** countLinesOfCode = 0; *// высчитаем при данном v количество строк*

**int** iter = pivot;

**while** (iter > 0) {

countLinesOfCode += iter;

iter /= k; *// сделали формулу рекурентной*

}

**if** (countLinesOfCode >= n) { *// проверяем написали ли мы необходимое количество строк кода*

v = pivot; *// если написали то ставим новый результат как наименьший*

rightPoint = pivot; *// смещаем правую точку отрезка. сужаем вдвое область поиска*

}

**else** leftPoint = pivot + 1; *// если с этим числом не получилось то перемистимся на правую часть отрезка*

*// для это сдвинем левую точку на половину отрезка*

}

System.out.println(v);

}

}

Результат тестирования:



**Задача 3:**

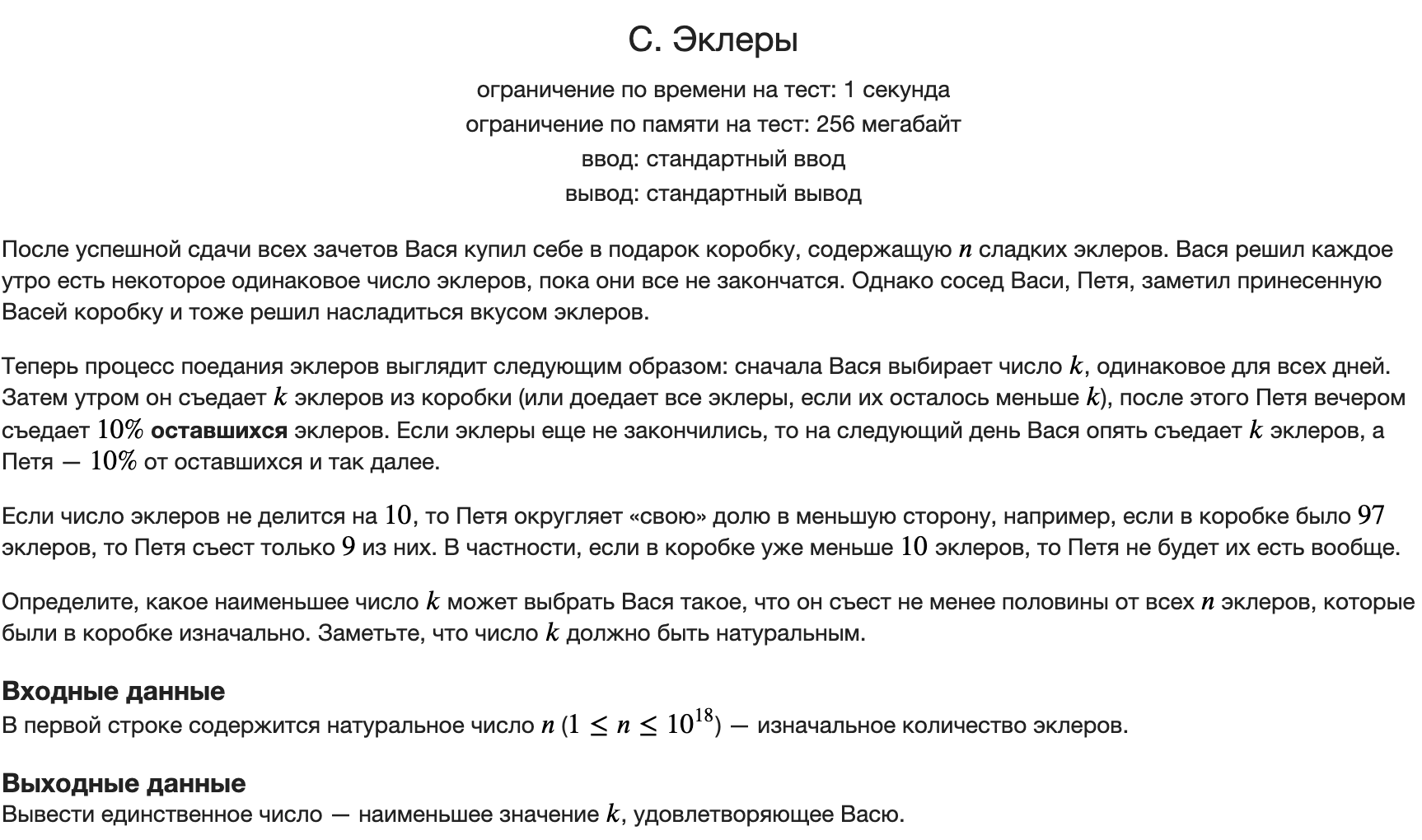
Условие задачи:  
Сложность:

Код задачи и пояснение к решению:

Результат тестирования:

**Задача 4:**

Условие задачи: <https://codeforces.com/problemset/problem/991/C>  
Сложность: 1500



Код задачи и пояснение к решению:

Результат тестирования:

**Вывод:**