

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

## Алгоритмы и структуры данных

Отчёт по лабораторной работе №1 (2025)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

## 1. Цель работы

Решить задачу №2025 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=2025>

## 2. Задача

### Условие

Бокс, каратэ, самбо... Классические боевые единоборства пресытили аудиторию. Поэтому известный спортивный канал запускает новый формат соревнований, основанный на традиционной русской забаве — боях стенка на стенку. В соревновании могут участвовать от двух до  $k$  команд, каждая из которых будет соперничать с остальными. Всего в соревновании примут участие  $n$  бойцов. Перед началом боя они должны разделиться на команды, каждый боец должен войти ровно в одну команду. За время боя два бойца сразятся, если они состоят в разных командах. Организаторы считают, что популярность соревнований будет тем выше, чем больше будет количество схваток между бойцами. Помогите распределить бойцов по командам так, чтобы максимизировать количество схваток между бойцами, и выведите это количество.

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

### Исходные данные

В первой строке дано количество тестов  $T$  ( $1 \leq T \leq 10$ ). В следующих  $T$  строках перечислены тесты. В каждой из них записаны целые числа  $n$  и  $k$  через пробел ( $2 \leq k \leq n \leq 104$ ).

### Результат

Для каждого теста в отдельной строке выведите одно целое число — ответ на задачу.

### Пример

Исходные данные		Результат
3		12
6	3	10
5	5	4
4 2		

## 3. Материалы работы

### 3.1. Вариант 1

#### 3.1.1. Объяснение алгоритма

Максимальное число боёв может достигаться тогда, когда распределение участников по командам будет максимально равномерным, чтобы каждый из участников смог сразиться с максимальным количеством противников. В случае, если участников нельзя разделить поровну, число участников в команде должно равняться  $n/k$  или  $n/k+1$ . Число боёв вчитается циклом, исходя из мысли, что, условно, если каждый участник команды  $A$  сразился со всеми участниками всех других команд, то команда  $A$  больше не участвует в поединках, значит её можно «снять с состязаний».

### 3.1.2. Код программы.

```
1. #include <iostream>
2.
3. using namespace std;
4.
5. int main(int argc, char const * argv[]) {
6.
7.     int t, n, k;
8.     int team_members, team_members_r, remainder, sum;
9.     cin >> t;
10.
11.     for (int i = 0; i < t; i++) {
12.         cin >> n >> k;
13.         team_members = n / k;
14.         team_members_r = team_members + 1;
15.         remainder = n % k;
16.         sum = 0;
17.         for (int j = 0; j < k - remainder; j++) {
18.             n -= team_members;
19.             sum += n * team_members;
20.         }
21.         for (int j = 0; j < remainder; j++) {
22.             n -= team_members_r;
23.             sum += n * team_members_r;
24.         }
25.         cout << sum << endl;
26.     }
27.     return 0;
```

## 3.2. Вариант 2

### 3.2.1. Объяснение алгоритма

В процессе работы над отчётом появилась идея оптимизации алгоритма и избавления от циклов. Если рассматривать всех участников, как вершины графа, то бои можно представить в виде рёбер графа. Максимальное число боёв – все вершины графа соединены. В таком случае граф будет являться полным, и число рёбер вычисляется по формуле  $n*(n-1)/2$ , где  $n$  – число участников. Чтобы исключить бои внутри команды, рассмотрим каждую команду как отдельный граф и вычтем из общего числа боёв число возможных боёв внутри команды.

### 3.2.2. Код программы.

```
1. #include <iostream>
2. using namespace std;
3.
4. int main(int argc, char const *argv[]) {
5.
6.     int t,n,k;
7.     int team_members,team_members_r,remainder,sum;
8.     cin >> t;
9.
10.    for (int i=0; i<t; i++){
11.        cin >> n >> k;
12.        team_members = n/k;
13.        remainder = n%k;
14.        sum = n*(n-1)/2 - (k-remainder)*team_members*(team_members-1)/2 -
            (remainder)*team_members*(team_members+1)/2;
15.        cout << sum << endl;
16.
17.    }
18.    return 0;
19. }
```

## 4. Результат выполнения и ссылка на репозиторий Github

### Результаты проверки решений

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
9239952	13:51:51 22 фев 2021	<a href="#">Danil Marukhlenko</a>	<a href="#">2025. Стенка на стенку</a>	Visual C++ 2019	Accepted		0.015	232 КБ
9239950	13:51:23 22 фев 2021	<a href="#">Danil Marukhlenko</a>	<a href="#">2025. Стенка на стенку</a>	Visual C++ 2019	Accepted		0.015	236 КБ

[https://github.com/japersik/algorithms\\_and\\_data\\_structures/tree/master/task\\_1\\_2025](https://github.com/japersik/algorithms_and_data_structures/tree/master/task_1_2025)



## 5. Вывод

Работа выполнена, задача была решена «в лоб», сложением всех возможных боёв, и более оптимально, с использованием ранее полученных знаний из других дисциплин.