**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по задачам №1005, №1296, №2025

по дисциплине «**Алгоритмам и структурам данных**»

Автор: Братушка Н. И.

Факультет: СУиР

Группа: R3238

Поток: АиСД 2.2

Преподаватель: Тропченко А. А.

Изображение выглядит как черный, темнота

Автоматически созданное описание

Санкт-Петербург 2024

Содержание

[Задача 1005. Куча Камней 3](#_Toc159513639)

[Текст задачи 3](#_Toc159513640)

[**Исходные данные** 3](#_Toc159513641)

[**Результат** 3](#_Toc159513642)

[**Пример** 3](#_Toc159513643)

[Программа для решения задачи 3](#_Toc159513644)

[Результат на сайте Timus 4](#_Toc159513645)

[Задача 1296. Гиперпереход 5](#_Toc159513646)

[Текст задачи 5](#_Toc159513647)

[**Исходные данные** 5](#_Toc159513648)

[**Результат** 5](#_Toc159513649)

[**Примеры** 5](#_Toc159513650)

[Программа для решения задачи 6](#_Toc159513651)

[Результат на сайте Timus 6](#_Toc159513652)

[Задача 2025. Стенка на стенку 7](#_Toc159513653)

[Текст задачи 7](#_Toc159513654)

[**Исходные данные** 7](#_Toc159513655)

[**Результат** 7](#_Toc159513656)

[**Пример** 7](#_Toc159513657)

[Программа для решения задачи 8](#_Toc159513658)

[Результат на сайте Timus 9](#_Toc159513659)

# **Задача 1005. Куча Камней**

## **Текст задачи**

У вас есть несколько камней известного веса *w*1, …, *wn*. Напишите программу, которая распределит камни в две кучи так, что разность весов этих двух куч будет минимальной.

**Исходные данные**

Ввод содержит количество камней *n* (1 ≤ *n* ≤ 20) и веса камней *w*1, …, *wn* (1 ≤ *wi* ≤ 100 000) — целые, разделённые пробельными символами.

**Результат**

Ваша программа должна вывести одно число — минимальную разность весов двух куч.

**Пример**

|  |  |
| --- | --- |
| **исходные данные** | **результат** |
| 5  5 8 13 27 14 | 3 |

**Источник задачи:**Чемпионат УрГУ 1997

## **Программа для решения задачи**

**A computer screen shot of a program code

Description automatically generated**

Для решения данной задачи необходимо распределить камни по двум кучам так, чтобы разница весов двух куч минимальна. Данную задачу можно решить с помощью рекурсии. Для начала мы считываем число камней и массив, в котором содержатся значения весов камней. Далее мы передаем данные в функцию, которая рекурсивно вызывает себя, каждый раз прибавляя камень к одной из куч. Когда мы распределим все камни, функция вычислит разность весов, и если она будет меньше предыдущего значения, то мы присваиваем переменной разности новое значение.

Например, если у нас имеется 4 камня, то функция рассмотрит следующие комбинации:

4 0

3 1

2 2

1 3

0 4

## **Результат на сайте Timus**

**A screenshot of a computer

Description automatically generated**

# **Задача 1296. Гиперпереход**

## **Текст задачи**

Гиперпереход, открытый ещё в начале XXI-го века, и сейчас остаётся основным способом перемещения на расстояния до сотен тысяч парсеков. Но совсем недавно физиками открыто новое явление. Оказывается, длительностью альфа-фазы перехода можно легко управлять. Корабль, находящийся в альфа-фазе перехода, накапливает гравитационный потенциал. Чем больше накопленный гравитационный потенциал корабля, тем меньше энергии потребуется ему на прыжок сквозь пространство. Ваша цель — написать программу, которая позволит кораблю за счёт выбора времени начала альфа-фазы и её длительности накопить максимальный гравитационный потенциал.

В самой грубой модели грави-интенсивность — это последовательность целых чисел *pi*. Будем считать, что если альфа-фаза началась в момент *i* и закончилась в момент *j*, то накопленный в течение альфа-фазы потенциал — это сумма всех чисел, стоящих в последовательности на местах от *i* до *j*.

**Исходные данные**

В первой строке записано целое число *N* — длина последовательности, отвечающей за грави-интенсивность (0 ≤ *N* ≤ 60000). Далее идут *N* строк, в каждой записано целое число *pi* (−30000 ≤ *pi* ≤ 30000).

**Результат**

Выведите максимальный гравитационный потенциал, который может накопить корабль в альфа-фазе прыжка. Считается, что потенциал корабля в начальный момент времени равен нулю.

**Примеры**

|  |  |
| --- | --- |
| **исходные данные** | **результат** |
| 10  31  -41  59  26  -53  58  97  -93  -23  84 | 187 |
| 3  -1  -5  -6 | 0 |

**Автор задачи:**Ден Расковалов  
**Источник задачи:**IX Открытое командное соревнование школьников по программированию (13.03.2004)

## **Программа для решения задачи**

**A computer screen shot of a program code

Description automatically generated**

В данной задачи найти максимальным потенциал, который может накопить корабль. То есть нам нужно найти подпоследовательность в последовательности чисел, сумма которой будет наибольшей. Для этого считываем число элементов последовательности и запускаем цикл, который выполнится столько раз, сколько и число элементов. Каждый раз мы будем считывать новое число и прибавлять его к переменной probable (сумма подпоследовательности). Далее мы будем проверять знак значения переменной: если значение отрицательно, то это значит, что нет никакого смысла нам брать эту подпоследовательность и мы обнуляем переменную для следующей подпоследовательности; иначе – проверяем, стала ли сумма подпоследовательности больше потенциала (по условию, начальное значение равняется нулю). Если предположение оказалось верным, то мы присваиваем потенциалу новое значения. После выполнения цикла выводим ответ.

## **Результат на сайте Timus**

**A screenshot of a computer

Description automatically generated**

# **Задача 2025. Стенка на стенку**

## **Текст задачи**

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.

**Исходные данные**

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

**Результат**

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

**Пример**

|  |  |
| --- | --- |
| **исходные данные** | **результат** |
| 3  6 3  5 5  4 2 | 12  10  4 |

**Автор задачи:**Алексей Данилюк  
**Источник задачи:**Уральская региональная командная олимпиада по программированию 2014

## **Программа для решения задачи**

**A computer screen with text on it

Description automatically generated**

Для решения данной задачи необходимо посчитать максимальное число боев, которое можно провести с заданным числом команд и бойцов. В случае, когда число бойцов делится нацело на количество команд, нам необходимо:

1. Вычислить число бойцов в других командах (число боев 1 бойца команды)
2. Умножить данное число на количество бойцов (получим удвоенное число всех бойцов, так как мы дважды считаем один и тот же бой для каждого бойца)
3. Поделить полученное значение на 2 и вывести итоговый ответ

В случае, если число бойцов не делится нацело на количество команд, мы:

1. Вычисляем количество бойцов при равномерном распределении (если у нас 8 бойцов и 3 команды, то мы получим 2 бойца на команду) и количество оставшихся, «лишних» бойцов.
2. Вычисляем количество боев между распределенными бойцами согласно вышеупомянутому алгоритму
3. Вычисляем количество «лишних» бойцов с изначально распределёнными бойцами (при неудобных числах, во всех командах, кроме одной, может быть добавлен только один боец)
4. Вычисляем количество боев «лишних» бойцов между собой вышеупомянутым алгоритмом
5. Складываем полученные значения и выводим ответ

## **Результат на сайте Timus**

**A screenshot of a computer

Description automatically generated**