|  |  |
| --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  РОССИЙСКОЙ ФЕДЕРАЦИИ  Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | |
|  | Институт компьютерных наук и технологий |
| ОТЧЕТ  по лабораторной работе **№6**  по дисциплине Языки программирования  «**ООП**» | |
|  | Работу выполнила студентка  Группы ПМИ 1-2, курс 2  Института компьютерных наук и технологий  **Арланова Анна Андреевна** |
|  | Работу принял:  ассистент кафедры Информационных технологий  Ракина В.Д. |
| Пермь, 2023 | |

Оглавление

[1 Постановка задачи 3](#_Toc128521975)

[2 Проектирование 4](#_Toc128521976)

[3 Тестирование 6](#_Toc128521977)

[4 Приложение 13](#_Toc128521978)

# Постановка задачи

1. Бинарные файлы, содержащие числовые данные (исходный файл заполнить случайными данными, заполнение организовать отдельным методом). Вычислить произведение нечетных отрицательных компонент файла.
2. Бинарные файлы, содержащие числовые данные (исходный файл заполнить случайными данными, заполнение организовать отдельным методом. Скопировать элементы заданного файла в квадратную матрицу размером n×n (если элементов файла недостает, заполнить оставшиеся матрицы нулями). Поменять местами в каждом столбце минимальный и максимальный элементы.
3. Бинарные файлы, содержащие величины типа struct (заполнение исходного файла организовать отдельным методом). Информация о багаже пассажира описывается массивом, где каждый элемент содержит название единицы багажа (чемодан, сумка, коробка и т.д.) и ее массу. Дан файл, содержащий сведения о багаже нескольких пассажиров. Найти число пассажиров, имеющих более двух единиц багажа и число пассажиров, количество единиц багажа которых превосходит среднее число единиц багажа.
4. Решить задачу с использованием структуры «текстовый файл» (в файле хранятся целые числа по одному в строке). В файле найти сумму квадратов элементов.
5. Решить задачу с использованием структуры «текстовый файл» (в файле хранятся целые числа по несколько в строке). Вычислить произведение элементов.
6. Решить задачу с использованием структуры «текстовый файл» (в файле хранится текст). Переписать в другой файл строки, имеющие заданную длину m.

# Проектирование

Разделила программу на 3 файла:

1. FileFunction.cs - все необходимые методы для решения задач
2. UserInput.cs – проверка корректности пользовательского ввода
3. Program.cs – вывод менюшки и выбор задания
4. Passenger.cs – структура пассажира

До выбора задания, программа создает бинарный файл и заполняет его случайными значениями. Пользователь должен задать количество элементов в файле и диапозон. После выводится, то что сгенерировалось в данном файле. Так же благодаря менюшке, пользователь может пересоздать бинарный файл и посмотреть на его содержимое.

## Произведение нечетных отрицательных компонент файла

Берется бинарный файл, что создался в меню пользователем. Считывается число с файла по заданному пути. Проверяет является ли оно меньше нуля и не делится ли на 2. В хорошем исходе итоговое произведение умножается на это число. Если ни разу не перемножилось на число, то выводится ноль.

## Поменять местами в каждом столбце минимальный и максимальный элементы

Берется тот же бинарный файл, что создался в меню пользователем. Запрашивается ввод размерности матрицы от пользователя, если все хорошо, то создается матрица с такой размерностью. После она заполняется данными из бинарного файла по строчно, если данных не хватает, заполняется нулями. Выводится заполненная матрица. Затем применяется метод на перестановку местами в каждом столбце минимального и максимального элемента. Суть его такая: происходит перебор каждого столбца матрицы. Для каждого столбца определяются индексы минимального и максимального элементов в этом столбце соответственно. Для этого происходит перебор элементов в столбце и сравнение их значений с ранее найденным минимальным и максимальным значениями. После нахождения минимального и максимального элементов в столбце происходит их замена местами, чтобы минимальный элемент стал максимальным, а максимальный стал минимальным. В конце, выводится измененная матрица.

## Число пассажиров, имеющих более двух единиц багажа, количество единиц багажа которых превосходит среднее число единиц багажа

Создается массив пяти элементов из структуры Passenger, которая описывается в отдельном файле Passenger.cs. В структуре два поля, первое - имя пассажира, представленное строкой. Второе поле - список багажа, представленное как не странно списком из кортежа строки и целого числа, где строка название багажа, а число – вес. Также присутвует красивый вывод для этой структуры. После программа выводит, то что находится в массиве. После этого пытаемся записать наш массив в бинарный файл при помощи сериализации. Если все успешно, выводится сообщение о том, где был создан файл. Далее считываем бинарный файл и преобразуем его содержимое в массив пассажиров при помощи десериализации. В случае успеха выводим ссобщение откуда мы считали. И также выводим, то что мы записали в массив, чтобы проверить, что все правильно. Далее проходимся по массиву и подсчитываем количество пассажиров, у которых в списке багажа более чем 2 элемента. Также подсчитываем общее количество багажа у всех пассаижров. Вычисляем среднее значение количества багажа на пассажира. Проходимя по массиву и подсчитываем количество пассажиров у которого количество багажа превышает это среднее значение. В конце выводим информацию, которую подсчитали.

## Сумма квадратов элементов

Создается текстовый файл, где в каждой строчке будет по одному числу. Создается аналогично бинарному, то есть от пользователя запрашивается количество элементов и их диапозон. Выводится информация, где был сохранен файл. Далее считываем с этого файла данные и паралельно подсчитываем сумму квадратов элементов. Выводим соотвествующую информацию.

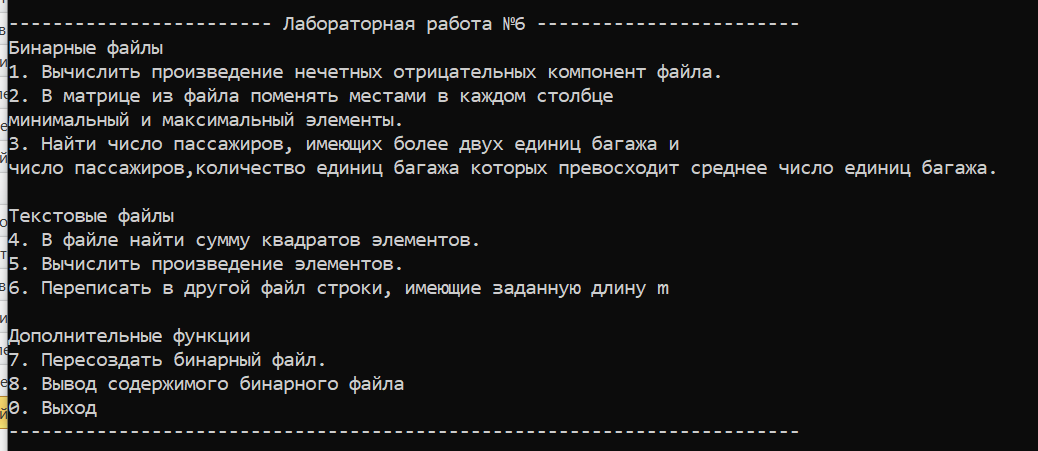
## Произведение элементов

Создается текстовый файл, где в каждой строчке может быть не одно число, разделенное пробелом. Создается аналогично бинарному, то есть от пользователя запрашивается количество элементов и их диапозон. С некоторым шансом может записаться в файл перенос строки, таким образом, мы можем рандомно создать файл где в первой строчке, например, 4 элемента, на второй – 2, а на третьей 5 итд. Выводится информация, где был сохранен файл. Далее мы считываем информацию с этого файла построчно и разделяем эту строку по пробелу и переумножаем все считанные элементы. Выводим информацию о найденном произведении.

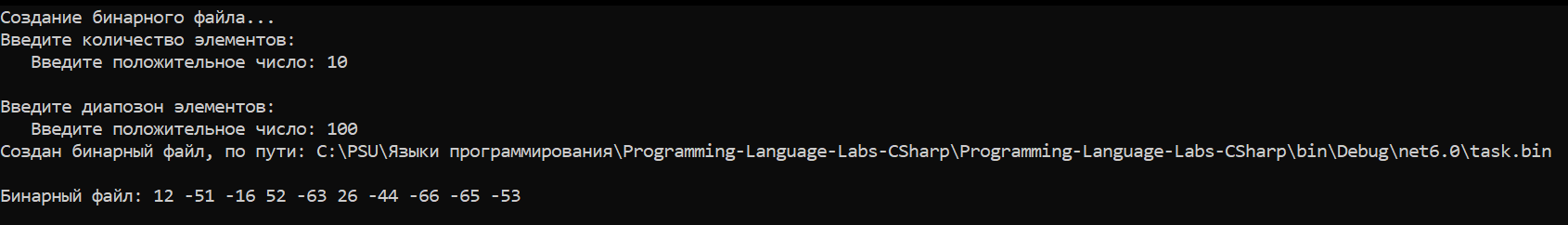
## Переписать в другой файл строки, имеющие заданную длину m

Создается текстовый файл, где пользователь сам может написать содержимое в несколько строк. Чтобы закончить вывод надо вести пустую строку. Выводится информация, куда сохранился файл. Далее от пользователя запрашивается целое число, длина строки. Создается другой текстовый файл, куда по итогу записываются строки с длинной, совпадающие тому, что вел пользователь. Выводится информация, где был создан новый файл.

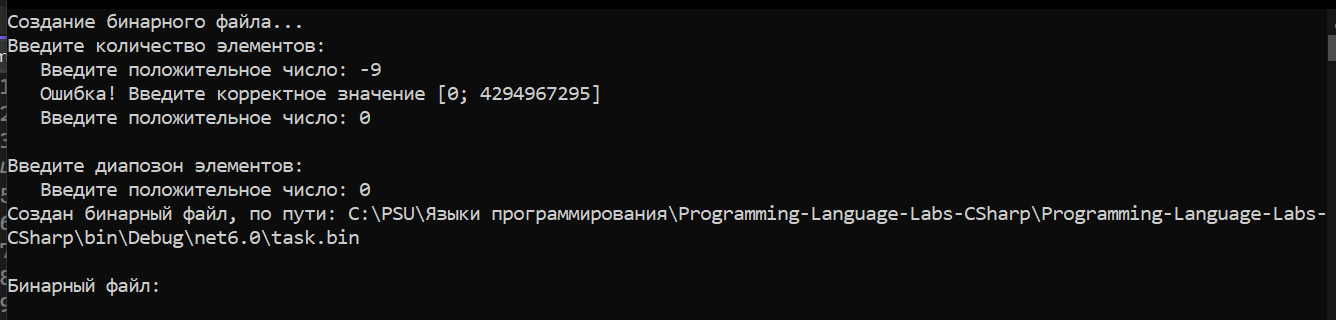
# Тестирование



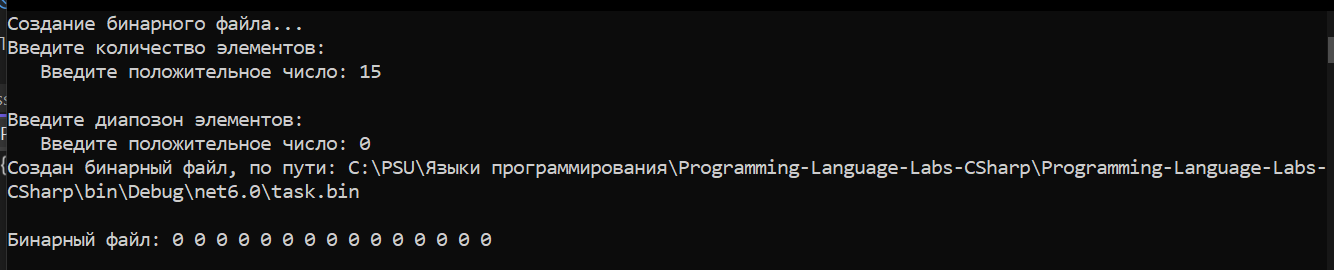
## Создание бинарного файла



1 тест

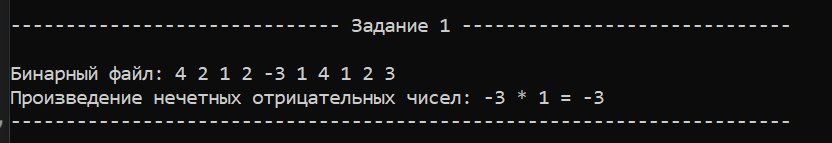


2 тест

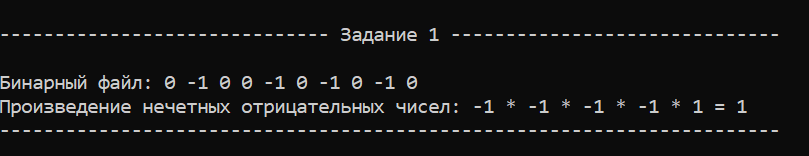


3 тест

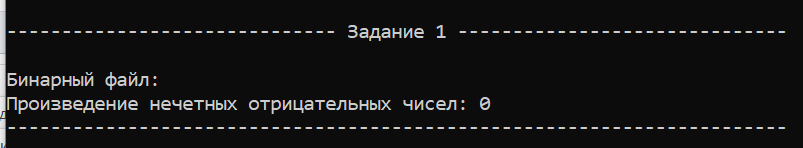
## Произведение нечетных отрицательных компонент файла



1 тест

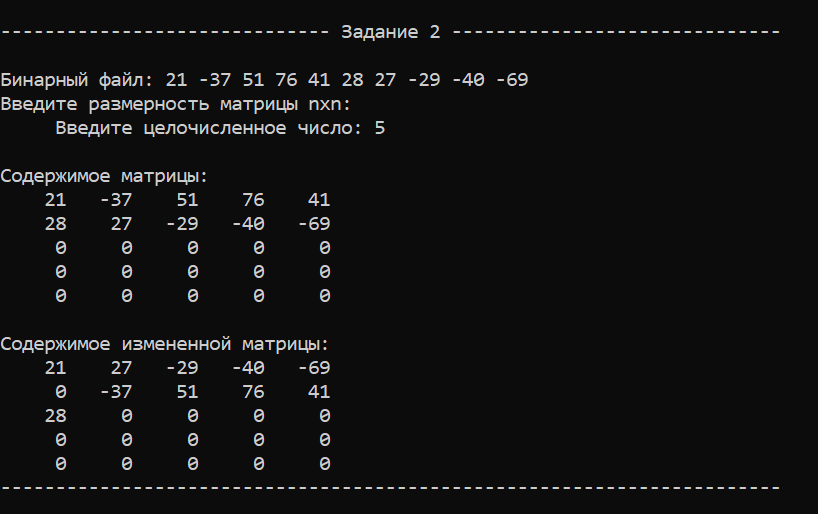


2 тест

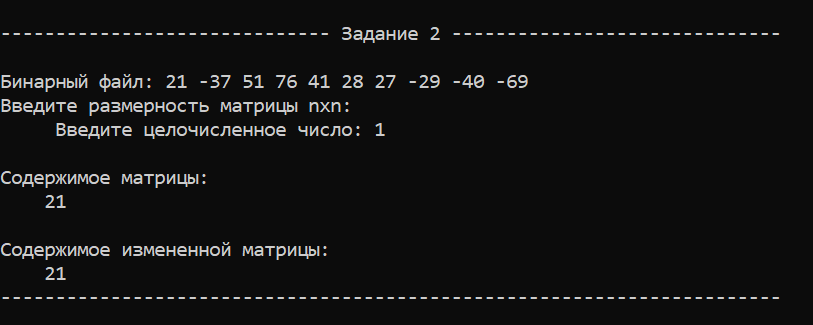


3 тест

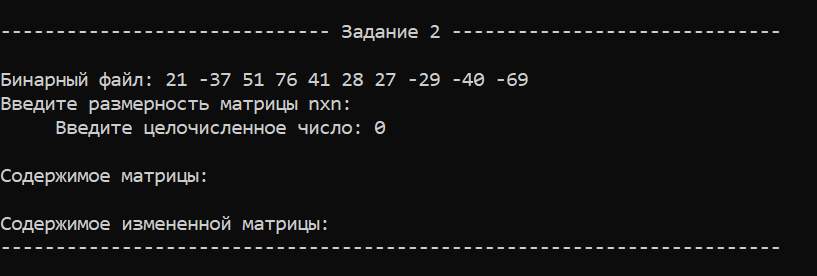
## Поменять местами в каждом столбце минимальный и максимальный элементы



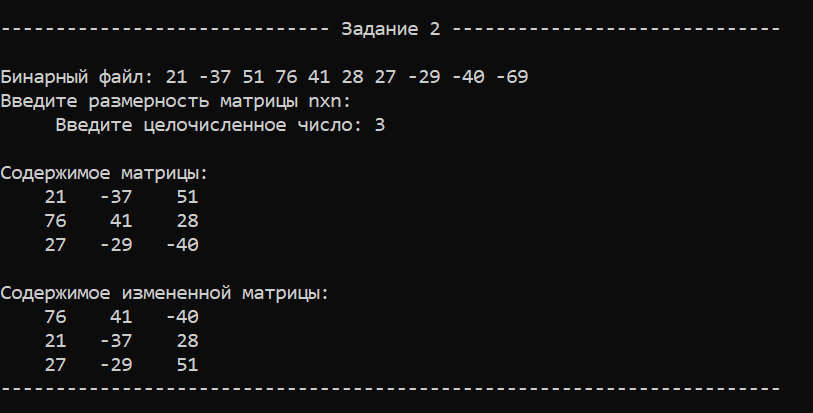
1 тест



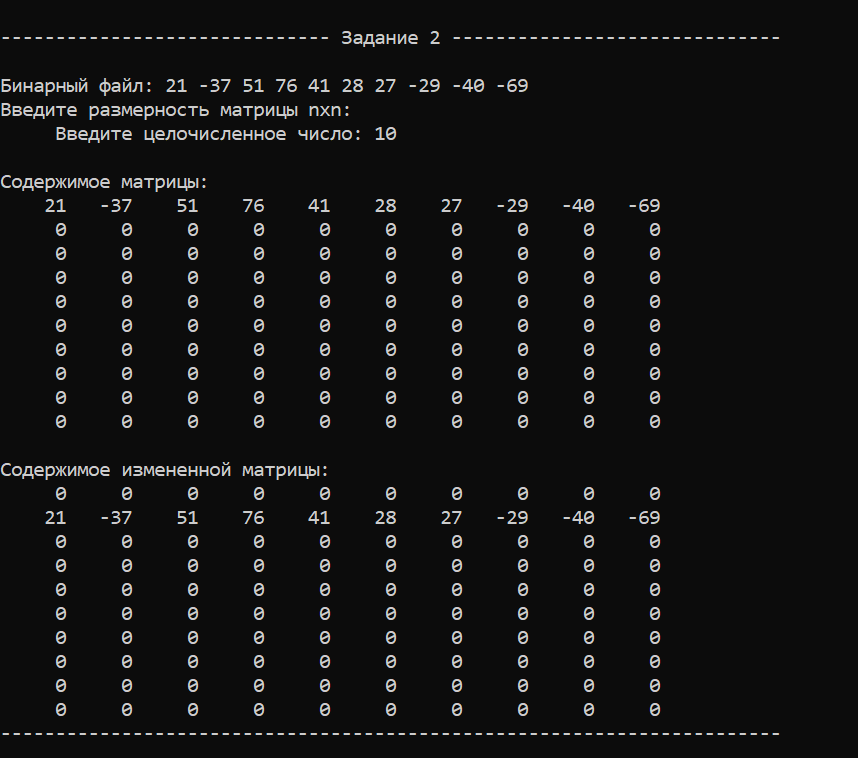
2 тест



3 тест

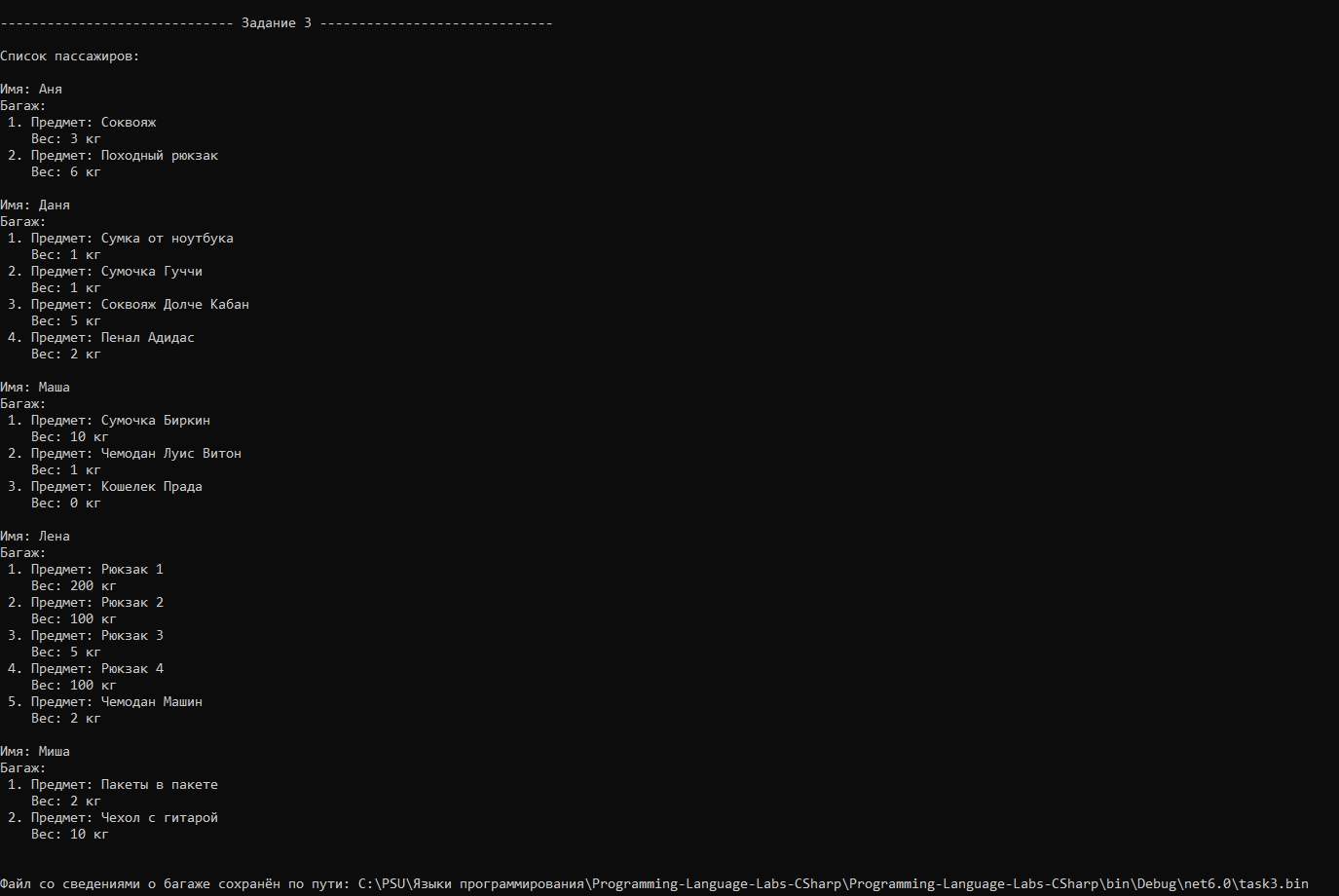


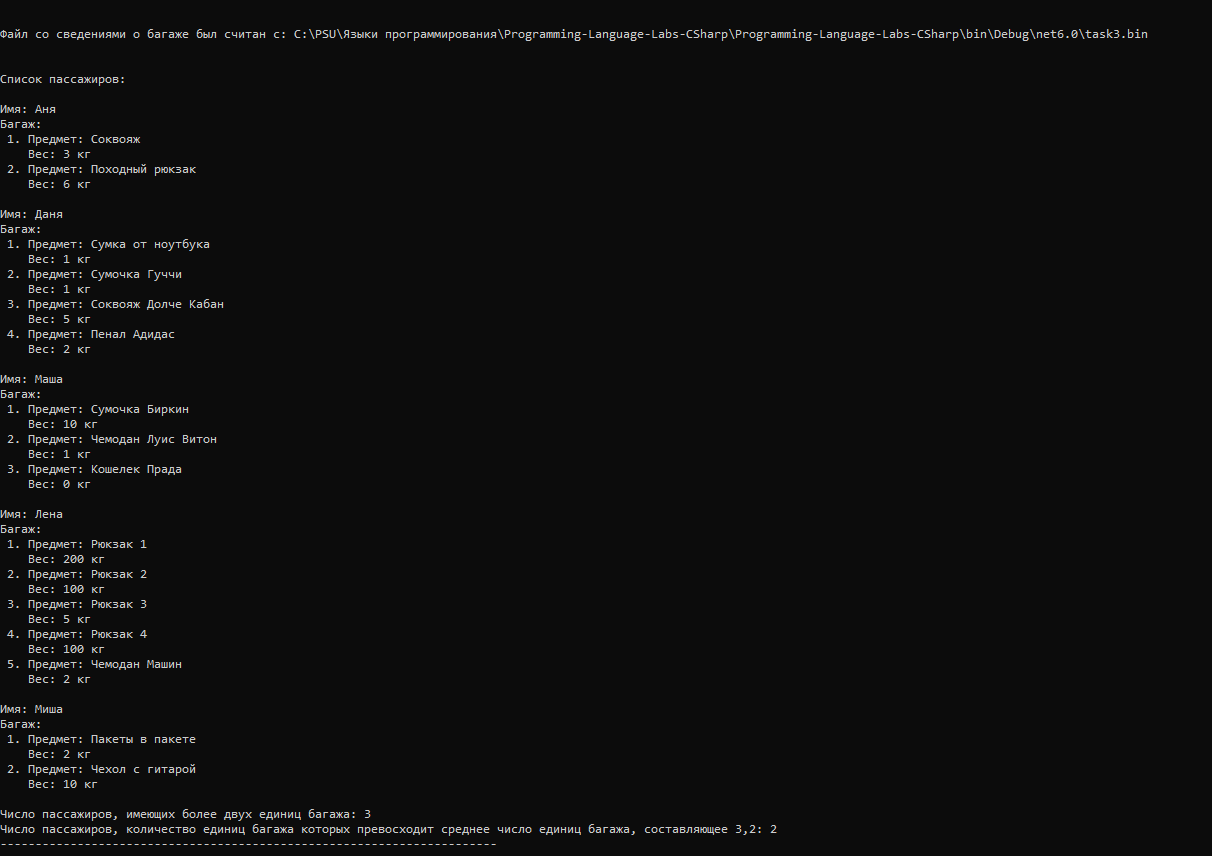
4 тест



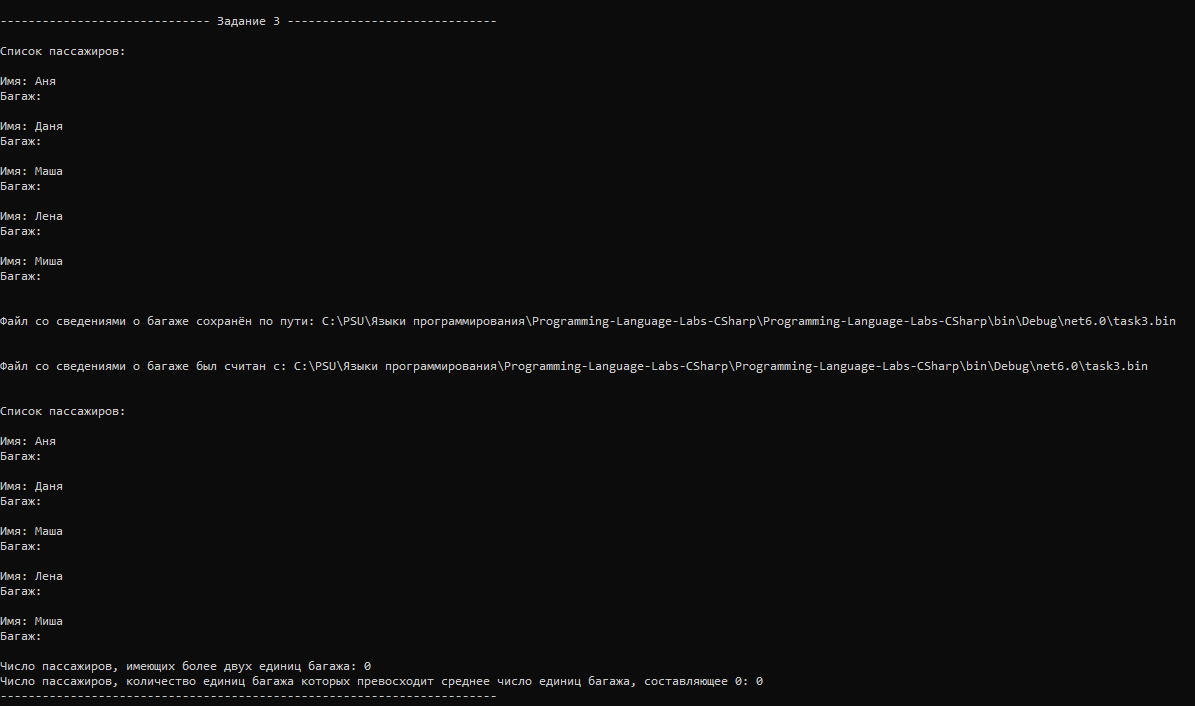
5 тест

## Число пассажиров, имеющих более двух единиц багажа, количество единиц багажа которых превосходит среднее число единиц багажа



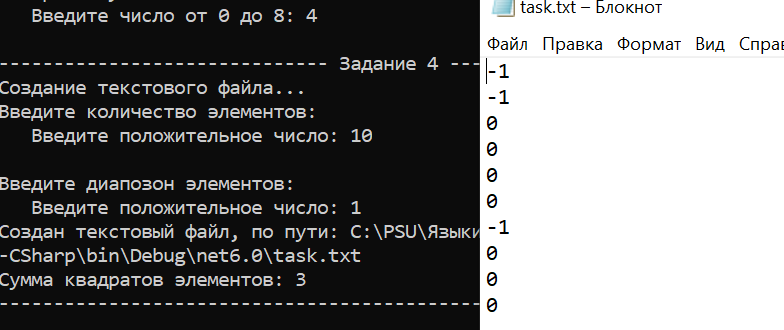


1 тест

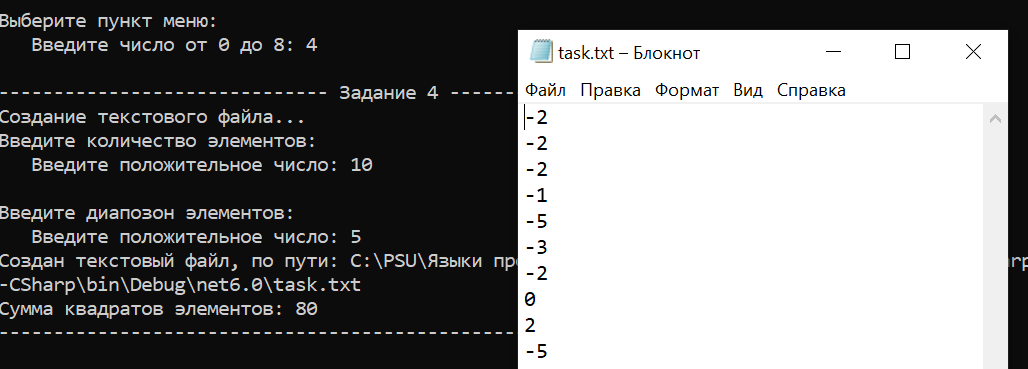


2 тест

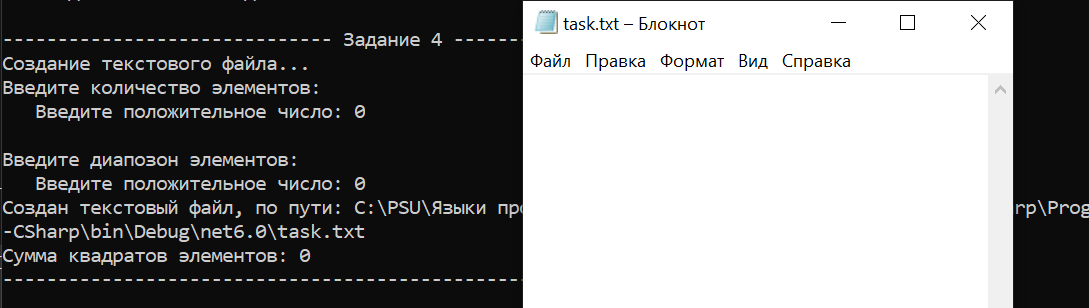
## Сумма квадратов элементов



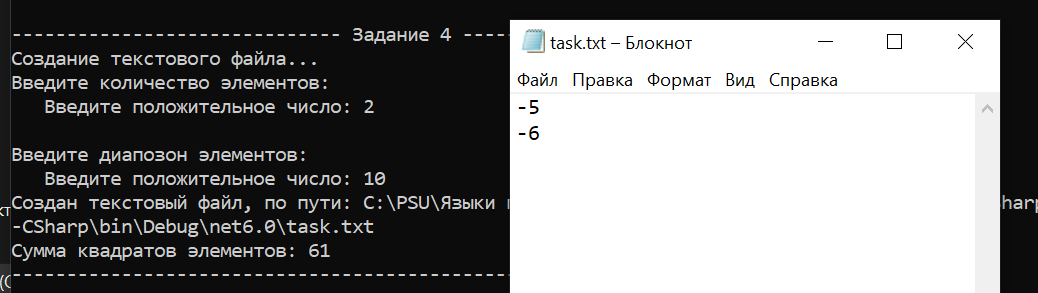
1 тест



2 тест

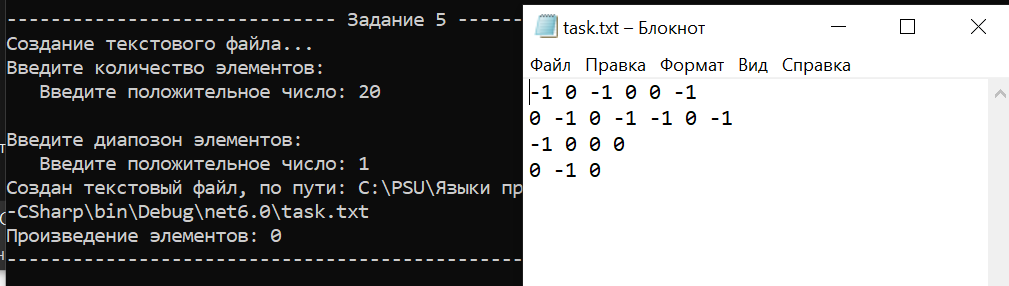


3 тест

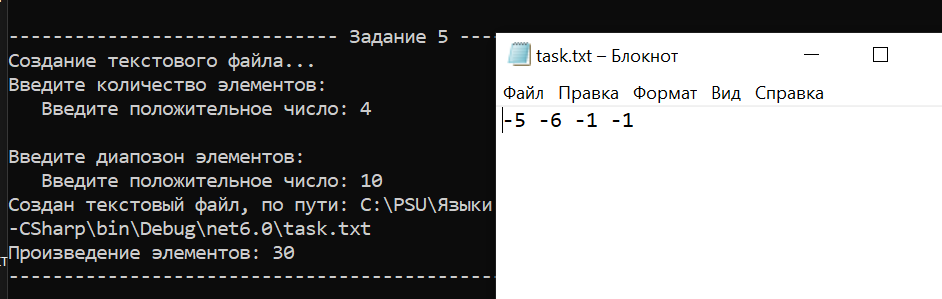


4 тест

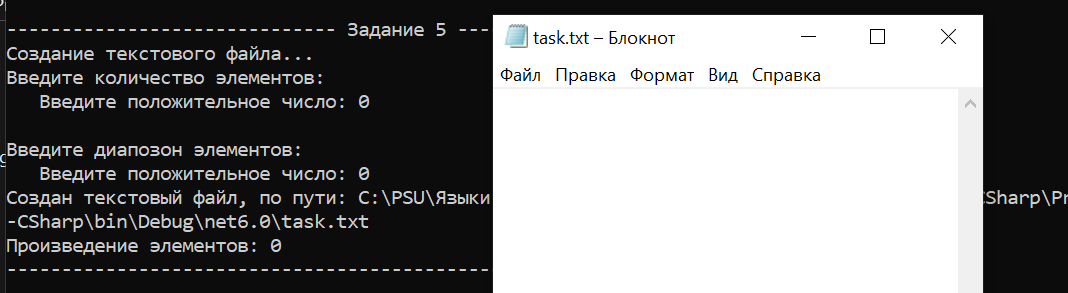
## Произведение элементов



1 тест

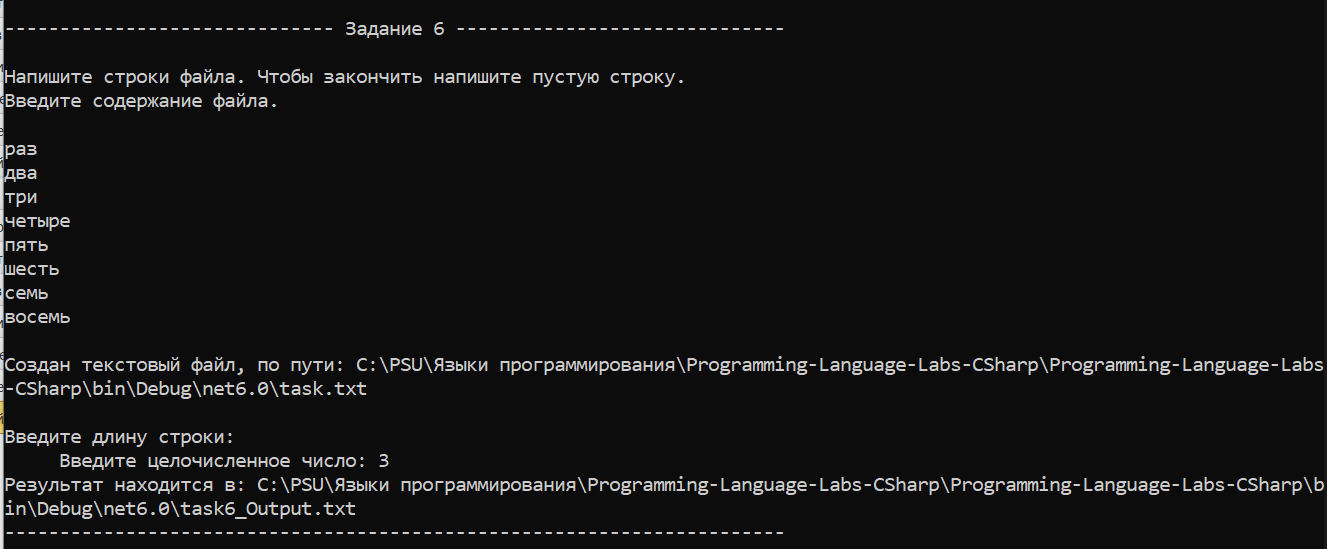


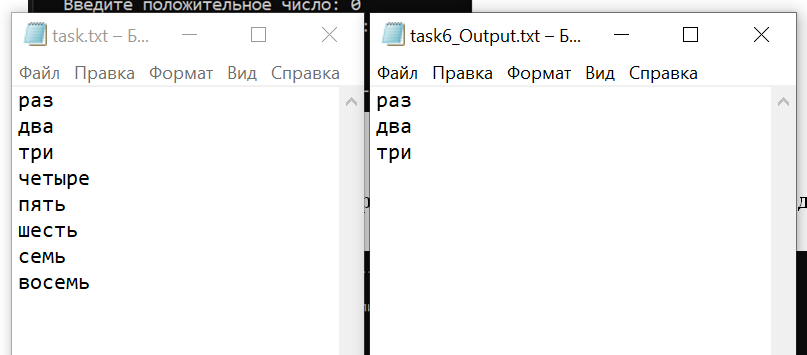
2 тест



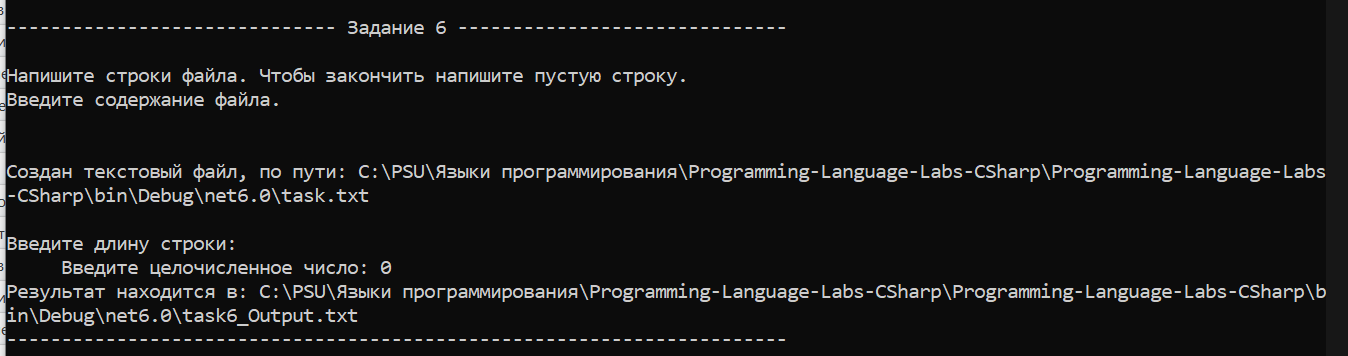
3 тест

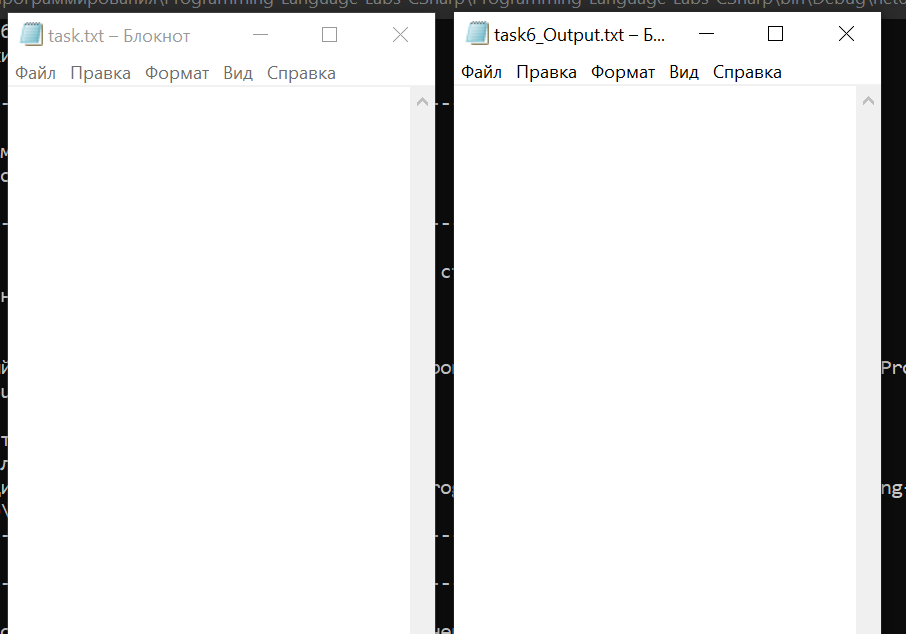
## Переписать в другой файл строки, имеющие заданную длину m





1 тест





2 тест