**Часть 1.**

**Задание 0.**

Дан массив с int переменными, определить количество пар, разность которых эквивалентна заданному с консоли target значению.

Например, дан массив чисел [1,2,3,4] и введенное с консоли target значение 1, у нас есть три пары, которые подходят под заданное условие: 2-1=1, 3-2=1, 4-3=1.

Нужно дописать код в pairs функции, которая принимает массив int значений и target int переменную, которая бы возвращала количество пар с заданной разностью.

**Параметры:**

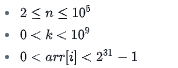
* k: int, target переменная, разница
* arr: массив int значений

**Формат ввода:**

* Первая строка принимает 2 значения отделенных пробелом, размер массива arr и target значение. (n и k)
* Вторая строка содержит n отделенных друг от друга int значений массива arr.

**Ограничения:**

* каждое значение arr[i] должно быть уникальным



**Формат вывода:**

Целочисленное число, отображающее количество пар, разность которых равна k (target переменной)

**Пример ввода:**

5 2

1 5 3 4 2

**Пример вывода:**

3

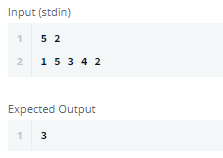
**Объяснение:**

Массив содержит 3 пары int переменных, разность которых равна 3:

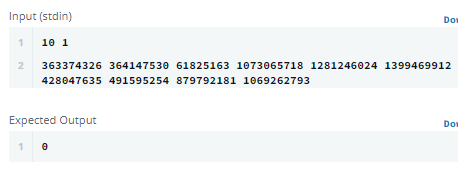
[5,3], [4,2], [3,1].

**Для проверки:**

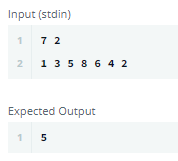
**1.**



**2.**



**3.**



**public** **class** Solution {

// Complete the pairs function below.

**static** **int** pairs(**int** k, **int**[] arr) {

**int** pairsCount = 0;

// write code here

**return** pairsCount;

}

**private** **static** **final** Scanner ***scanner*** = **new** Scanner(System.***in***);

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

String[] nk = ***scanner***.nextLine().split(" ");

**int** n = Integer.*parseInt*(nk[0]);

**int** k = Integer.*parseInt*(nk[1]);

**int**[] arr = **new** **int**[n];

String[] arrItems = ***scanner***.nextLine().split(" ");

**for** (**int** i = 0; i < n; i++) {

arr[i] = Integer.*parseInt*(arrItems[i]);

System.***out***.println(arr[i]);

}

**int** result = *pairs*(k, arr);

System.***out***.println(result);

***scanner***.close();

}

}

**Задание 1.**

Ввести две строки с консоли. Определить, является ли одна строка перестановкой символов другой строки. Учитываем регистр. Не использовать сортировку :)

Например:

1. “hello” и “hlleo” -> true
2. “hello” и “art” -> false

**Задание 2.**

Ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

**Задание 3.**

Условие здачи: ввести 2 слова, состоящие из четного числа букв. Получить слово состоящее из первой половины первого слова и второй половины второго слова.

**Задание 4.**

Напишите программу Deal.java, которая должна имитировать раздачу карт для игры в покер.

Программа получает число n, задаваемое с консоли пользователем, и раздает карты на n игроков (по 5 карт каждому) из рассортированной колоды.

Разделяйте пять карт, выданных каждому игроку, пустой строкой.

1. Необходимо реализовать программу, которая будет позволять вводить сделки купли-продажи товаров между двумя участниками. В каждой сделке может быть несколько товаров, сумма сделки рассчитывается из суммы всех товаров. Сумма каждого товара рассчитывается из его стоимости и количества. Программа должна позволить ввести информацию о сделках, сохранить ее в памяти и вывести на экран. Ввод информацию осуществляется с экрана (консоли). Количество дополнительной информацию о сделке, участнике, товаре придумать самому (больше одного поля для класса). Сделка должна содержать массив продуктов.

2. Добавить 3 типа продукта, наследоваться от Product, реализовать метод подсчета цены для каждого продукта. Добавить расчет цены со скидкой. (В каждом продукте своя скидка).

3. Написать класс, который проверял бы введенную дату на соответствие патерна: (использовать regexp)

a) dd/MM/yyyy

b) dd-MM-yyyy

4. С помощью SimpleDateFormat пропарсить данные, введеные с консоли, в зависимости от паттерна, к которому дата подходит.

Вывести строку в следующем формате:

День: <Day>

Месяц: <Month>

Year: <Year>

5. Создать интерфейс Validator, с методом validate, принимающим строку, возвращая true/false.

Написать класс AmericanPhoneValidator, который валидирует американский телефон(начинающийся на +1...)

Написать класс BelarusPhoneValidator, который валидирует белорусский телефон(начинающийся на +375...)

Написать класс EmailValidator в методе main, валидирующий email.

Все три класса должны реализовать интерфейс Validator.

6. Добавить в Deal меню, которое позволяет добавить продукт, удалить продукт из массива по названию продукта, подсчитать сделку(вывести инфу о сделке на экран).

7. Добавить в объект Deal поле deadline date, которую будет вводиться автоматически. (сегодня + 10 дней)

8. Добавить в User поле dateOfBirth, которое задается с консоли, проверяется по regexp на соответствие паттернам из 3 задания.

9. Добавить поле phone и email в User, которое будет проверятся нашими Validator из 5 задания. (проверка на белорусский телефон)