// *TODO: #1. Функция принимает массив пользователей. Исходные данные см. ниже. Возвращает объект, состоящий из двух полей: women и men. Данные* *поля являются массивами, в которых содержаться пользователи, подходящие по категории gender соответственно.* *Также, вместо двух полей first\_name и last\_name у каждого из объектов должнобыть поле fullName в котором будут объеденины убранные поля*

*(first\_name и last\_name). Количество пользователей может быть не ограничено.*

const ***users*** = *[  
 {* id: 1,  
 first\_name: 'Jeanette',  
 last\_name: 'Penddreth',  
 email: 'jpenddreth0@census.gov',  
 gender: 'Female',  
 ip\_address: '26.58.193.2',  
 *}*,  
 *{* id: 2,  
 first\_name: 'Petr',  
 last\_name: 'Jackson',  
 email: 'gfrediani1@senate.gov',  
 gender: 'Male',  
 ip\_address: '229.179.4.212',  
 *}*,  
*]*;  
/\* Пример результата выполнения:  
{  
 women: [  
 {  
 id: 1,  
 fullName: 'Jeanette Penddreth',  
 email: 'jpenddreth0@census.gov',  
 gender: 'Female',  
 ipAddress: '26.58.193.2',  
 }  
 ],  
 men: [  
 {  
 id: 2,  
 fullName: 'Petr Jackson',  
 email: 'gfrediani1@senate.gov',  
 gender: 'Male',  
 ip\_address: '229.179.4.212',  
 }  
 ]  
}  
 \*/

// *TODO: #2. Функция принимает массив объектов. Исходные данные представлены ниже. Возвращает массив пользователей, отсортированных по полю username в алфавитном порядке и возраст которых начинается от 21 года. Количество пользователей может быть не ограничено.*const ***initialData*** = *[  
 {* username: 'Maria',  
 age: 25,  
 *}*,  
 *{* username: 'Alex',  
 age: 21,  
 *}*,  
 *{* username: 'Oleg',  
 age: 14,  
 *}*,  
 *{* username: 'Dmitriy',  
 age: 35,  
 *}*,  
 *{* username: 'Oksana',  
 age: 72,  
 *}*,  
*]*;  
  
/\* Результат выполнения:  
  
[  
 { username: "Alex", age: 21 },  
 { username: "Dmitriy", age: 35 },  
 { username: "Maria", age: 25 },  
 { username: "Oksana", age: 72 }  
]  
 \*/

