

1. Фільтрування користувачів за віком та створення статистики:

Уявіть, що у вас є файл `users.js`, який містить масив об'єктів користувачів з інформацією про їх ім'я, вік та інші характеристики. Створіть програму, яка:

- Використовує користувачів віком від 18 до 60 років(обидва включно).
- Обчислює середній вік цих користувачів(цілим числом)
- Вивід має бути зрозумілий

Приклад використання:

Файл `users.js`:

```
const users = [  
  { name: "John", age: 25 },  
  { name: "Alice", age: 30 },  
  { name: "Bob", age: 20 },  
  { name: "Emma", age: 28 },  
  { name: "Michael", age: 42 },  
  { name: "Taras", age: 12 },  
  { name: "Ivan", age: 72 },  
  { name: "Anna", age: 32 },  
  { name: "Lee", age: 61 },  
  { name: "Late", age: 7 },  
];
```

Вивід: *// Середній вік користувачів віком від 18 до 60 років: 30.*

2. Об'єднання даних з різних джерел:

У вас є два масиви в файлі з даними про користувачів: `users.js` (те що в 1 завданні) та `users2`, кожен з яких містить масив об'єктів користувачів з різною інформацією. Створіть програму, яка:

- Об'єднує дані про користувачів з обох джерел у один масив користувачів.
- Виводить об'єднаний та впорядкований масив усіх імен користувачів в консоль.

Приклад використання:

Файл `users.js` (додаємо новий масив):

```
const users2 = [  
  { name: "Bob", age: 20 },  
  { name: "Emma", age: 28 },  
  { name: "Liam", age: 25 },  
  { name: "Olivia", age: 22 },  
  { name: "William", age: 30 },  
  { name: "Ava", age: 26 },  
  { name: "James", age: 29 },  
  { name: "Charlotte", age: 27 },  
  { name: "Noah", age: 23 },  
  { name: "Isabella", age: 24 },  
  { name: "Benjamin", age: 21 },  
  { name: "Sophia", age: 20 },  
  { name: "Mason", age: 29 },  
  { name: "Amelia", age: 25 },  
  { name: "Ethan", age: 22 },  
];
```

Вивід: // Відсортовані імена користувачів: [

```
'Alice', 'Amelia', 'Anna',  
'Ava', 'Benjamin', 'Bob',  
'Bob', 'Charlotte', 'Emma',  
'Emma', 'Ethan', 'Isabella',  
'Ivan', 'James', 'John',  
'Late', 'Lee', 'Liam',  
'Mason', 'Michael', 'Noah',  
'Olivia', 'Sophia', 'Taras',  
'William'  
]
```

3. Генерація випадкових паролів для користувачів:

У вас є два набори даних про користувачів(як в 2 завданні), які потрібно об'єднати та створити для них випадкові паролі. Кожному користувачеві потрібно призначити пароль, довжина якого буде обчислена за допомогою простого рівняння: *вік користувача поділити на довжину імені*(бажано заокруглювати через *floor*).

P.S. :Згенеруйте випадковий пароль, використовуючи набір символів нижче або інший:

"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789".

В кінці виведіть масив користувачів з їхніми випадковими паролями.

Вивід: // Користувачі з випадковими паролями: [

```
{ name: 'John', age: 25, password: 'WbTgGM' },  
{ name: 'Alice', age: 30, password: 'HuBvmk' },  
{ name: 'Bob', age: 20, password: 'qxm52N' },  
{ name: 'Emma', age: 28, password: 'VmvAJjw' },  
{ name: 'Michael', age: 42, password: 'cwF9mN' },  
{ name: 'Taras', age: 12, password: 'Uu' },  
{ name: 'Ivan', age: 72, password: 'Ht4FhF8B4ISOiaf2IV' },  
{ name: 'Anna', age: 32, password: 'uKAF1hAt' },  
{ name: 'Lee', age: 61, password: '6KmttGHBUEk5s54RDGNZ' },  
{ name: 'Late', age: 7, password: 'z' },  
{ name: 'Bob', age: 20, password: 'BGrd58' },  
{ name: 'Emma', age: 28, password: 'J17GdYB' },  
{ name: 'Liam', age: 25, password: 'C9BSZK' },  
{ name: 'Olivia', age: 22, password: 'lU6' },  
{ name: 'William', age: 30, password: 'P7al' },  
{ name: 'Ava', age: 26, password: 'P3z4jZGU' },  
{ name: 'James', age: 29, password: 'DweRf' },  
{ name: 'Charlotte', age: 27, password: 'NkZ' },  
{ name: 'Noah', age: 23, password: 'W6unS' },  
{ name: 'Isabella', age: 24, password: 'NXd' },  
{ name: 'Benjamin', age: 21, password: 'tR' },  
{ name: 'Sophia', age: 20, password: 'vSj' },  
{ name: 'Mason', age: 29, password: 'EtFrL' },  
{ name: 'Amelia', age: 25, password: 'qPQj' },  
{ name: 'Ethan', age: 22, password: 'UKKf' } ]
```