// No realizar la prueba en repl.it al hacerlo su respuesta queda disponible para otros postulantes

// No editar

const clients = [

{ id: 1, taxNumber: '86620855', name: 'HECTOR ACUÑA BOLAÑOS'},

{ id: 2, taxNumber: '7317855K', name: 'JESUS RODRIGUEZ ALVAREZ'},

{ id: 3, taxNumber: '73826497', name: 'ANDRES NADAL MOLINA'},

{ id: 4, taxNumber: '88587715', name: 'SALVADOR ARNEDO MANRIQUEZ'},

{ id: 5, taxNumber: '94020190', name: 'VICTOR MANUEL ROJAS LUCAS'},

{ id: 6, taxNumber: '99804238', name: 'MOHAMED FERRE SAMPER' }

];

const accounts = [

{ clientId: 6, bankId: 1, balance: 15000 },

{ clientId: 1, bankId: 3, balance: 18000 },

{ clientId: 5, bankId: 3, balance: 135000 },

{ clientId: 2, bankId: 2, balance: 5600 },

{ clientId: 3, bankId: 1, balance: 23000 },

{ clientId: 5, bankId: 2, balance: 15000 },

{ clientId: 3, bankId: 3, balance: 45900 },

{ clientId: 2, bankId: 3, balance: 19000 },

{ clientId: 4, bankId: 3, balance: 51000 },

{ clientId: 5, bankId: 1, balance: 89000 },

{ clientId: 1, bankId: 2, balance: 1600 },

{ clientId: 5, bankId: 3, balance: 37500 },

{ clientId: 6, bankId: 1, balance: 19200 },

{ clientId: 2, bankId: 3, balance: 10000 },

{ clientId: 3, bankId: 2, balance: 5400 },

{ clientId: 3, bankId: 1, balance: 9000 },

{ clientId: 4, bankId: 3, balance: 13500 },

{ clientId: 2, bankId: 1, balance: 38200 },

{ clientId: 5, bankId: 2, balance: 17000 },

{ clientId: 1, bankId: 3, balance: 1000 },

{ clientId: 5, bankId: 2, balance: 600 },

{ clientId: 6, bankId: 1, balance: 16200 },

{ clientId: 2, bankId: 2, balance: 10000 }

]

const banks = [

{ id: 1, name: 'SANTANDER' },

{ id: 2, name: 'CHILE' },

{ id: 3, name: 'ESTADO' }

];

/\*

SECCIÓN PROBLEMAS

- No promover la copia:

- No preguntar en StackOverflow, foros, o similares ya que estas preguntas/respuestas quedan disponibles a otros candidatos

- No subir a repositorios públicos (github, o similares)

- Otros sitios como codepen o editores de texto on-line (codepen, repl, o similares), dejan guardado el código, por lo que les pedimos tampoco usar editores on-line, la mejor forma de debuggear su código es usando un interprete de javascript como node y ejecutarlo de manera local

- Para nosotros es fácil detectar pruebas con copia, no pierda su tiempo intentando hacerlo

- Posteriormente, se evaluará conocimiento en es6 en entrevistas presenciales

- Las siguientes son preguntas básicas de Javascript y manejo de datos. Se evaluará eficiencia, ORDEN y claridad del código entregado.

- Se debe programar un algoritmo para cada método y que este retorne lo requerido.

- Debe usar nombres explicativos para sus variables.

- Usar sintaxis ES6.

- Los resultados son evaluados con un test automatizado, revise que sus retornos sean con la estructura de datos solicitada en cada pregunta.

- Métodos menos verbosos, DRY, y buenas prácticas en el código mejoran el puntaje final de su prueba

- Si necesita hacer supuestos que afecten las respuestas entregadas, por favor déjelos escritos en el cuerpo del correo cuando envíe su prueba (No en este archivo). Supuestos que contradigan lo solicitado, no serán considerados como válidos.

- Su prueba debe ejecutarse sin errores con: node nombre-apellido.js

\*/

// 0 Arreglo con los ids de clientes

function listClientsIds() {

return clients.map((client) => client.id);

};

// 1 Arreglo con los ids de clientes ordenados por rut

function listClientsIdsSortByTaxNumber() {

// CODE HERE

};

// 2 Arreglo con los nombres de cliente ordenados de mayor a menor por la suma TOTAL de los saldos de cada cliente en los bancos que participa.

function sortClientsTotalBalances() {

// CODE HERE

}

// 3 Objeto en que las claves sean los nombres de los bancos y los valores un arreglo con los ruts de sus clientes ordenados alfabeticamente por nombre.

function banksClientsTaxNumbers() {

// CODE HERE

}

// 4 Arreglo ordenado decrecientemente con los saldos de clientes que tengan más de 25.000 en el Banco SANTANDER

function richClientsBalances() {

// CODE HERE

}

// 5 Arreglo con ids de bancos ordenados crecientemente por la cantidad TOTAL de dinero que administran.

function banksRankingByTotalBalance() {

// CODE HERE

}

// 6 Objeto en que las claves sean los nombres de los bancos y los valores el número de clientes que solo tengan cuentas en ese banco.

function banksFidelity() {

// CODE HERE

}

// 7 Objeto en que las claves sean los nombres de los bancos y los valores el id de su cliente con menos dinero.

function banksPoorClients() {

// CODE HERE

}

// 8 Agregar nuevo cliente con datos ficticios a "clientes" y agregar una cuenta en el BANCO ESTADO con un saldo de 9000 para este nuevo empleado.

// Luego devolver el lugar que ocupa este cliente en el ranking de la pregunta 2.

// No modificar arreglos originales para no alterar las respuestas anteriores al correr la solución

function newClientRanking() {

// CODE HERE

}

// Impresión de soluciones. No modificar.

console.log('Pregunta 0');

console.log(listClientsIds());

console.log('Pregunta 1');

console.log(listClientsIdsSortByTaxNumber());

console.log('Pregunta 2');

console.log(sortClientsTotalBalances());

console.log('Pregunta 3');

console.log(banksClientsTaxNumbers());

console.log('Pregunta 4');

console.log(richClientsBalances());

console.log('Pregunta 5');

console.log(banksRankingByTotalBalance());

console.log('Pregunta 6');

console.log(banksFidelity());

console.log('Pregunta 7');

console.log(banksPoorClients());

console.log('Pregunta 8');

console.log(newClientRanking());