

**1 - [REFATORAÇÃO]** Dado um array de nomes, faça um programa que imprima na tela todos os nomes (na mesma linha).

**Os valores de entrada serão:**

Array;

**Exemplo de Entrada:**

['Angela', 'Rosa', 'Ticiana', 'Carla', 'Renata']

**Exemplo de Saída:**

Angela, Rosa, Ticiana, Carla, Renata

**2 -** Dado um array de números, faça um programa que teste se este array contém um valor específico. Tanto o valor quanto o array serão valores de entrada.

**Exemplo de Entrada:**

[1, 3, -12, 33, -1, 9, 110, -1168, 252, -15253, 2127]

9

**Exemplo de Saída:**

O valor foi encontrado no Array.

**3 - [REFATORAÇÃO]** Dado um array de números, faça um programa que imprima na tela somente os números positivos.

OBS: Para melhorar a resolução desse problema, crie funções que também imprima os números negativos, os pares, os ímpares e realize a contagem de cada um deles.

**Os valores de entrada serão:**

Array;

**Exemplo de Entrada:**

[1, -7, -23, 25, -19, 13, 10, -8, 52, -153, -127]

-----

[-5, -1467, 32, 28, 65, -2, -49, -63, 22, -13, 255 ]

**4 - [REFATORAÇÃO]** Crie um algoritmo que tem como entrada um array de notas, e imprime no console a média aritmética dessas notas.

OBS: Para melhorar a resolução desse problema, monte um programa que retorna a média aritmética independente do tamanho do array.

**Exemplo Entrada:**

[6, 8, 7]

**5 -** Escreva um programa para substituir os elementos pares pelo valor "X".

**Exemplo de Entrada:**

[1, -7, -23, 22, -19, 13, 10, -8, 52, -153, -127]

**Exemplo de Saída:**

[1, -7, -23, "X", -19, 13, "X", "X", "X", -153, -127]

**6 - [REFATORAÇÃO]** Dado um array de números como entrada, faça um programa que multiplique todos os valores por 10.

OBS: Para melhorar a resolução desse problema, monte um programa que recebe como parâmetro o valor pelo qual os valores do array devem ser multiplicados.

**Exemplo de Entrada:**

[1, 8, 6, 13, 7, 9, 15, 22, 356, 4, 3]

**7 -** Dado um array de entrada com o nome e as notas de avaliações dos alunos, retorne uma lista com os alunos que tiraram 0 em todas as provas.

**Exemplo de Entrada:**

```
[  
{nome: ,  
notas: [ ]  
}
```

**8 - [REFATORAÇÃO]** Dado um array de nomes, faça um programa que tenha como entrada um valor à ser pesquisado e retorne se o nome existe ou não no array.

**Exemplo de Entrada:**

```
[ 'Juliana', 'Pedro', 'Regina', 'João', 'Patrícia', 'Gabriel', 'Carlos', 'Carla' ]  
'Carla'
```

**Exemplo de Saída:**

O nome Carla existe no Array.

**9 - [REFATORAÇÃO]** Dado a lista de compra a seguir, faça um programa que imprime no console valor total. Atenção, o valor de cada item é unitário.

**Exemplo de Entrada:**

```
[  
  { nome: 'maçã', quantidade: 2, valor: 0.5},  
  { nome: 'alface', quantidade: 1, valor: 1.73},  
  { nome: 'Água 5L', quantidade: 2, valor: 5.99},  
  { nome: 'Pão Francês', quantidade: 8, valor: 0.63}  
]
```

**Exemplo de Saída:**

R\$ 19.75

**10** - Escreva um programa para encontrar os elementos comuns entre dois arrays

**Exemplo de Entrada:**

```
[1, -7, -23, 22, -19, 13, 10, -8, 52, -153, -127]  
[-5, -19, 32, 28, 65, -2, -49, -63, 22, 13, 255 ]
```

**Exemplo de Saída:**

```
[22, -19, 13]
```

**11 - [REFATORAÇÃO]** Crie um algoritmo que tem como entrada um array de strings e trata essas string substituindo os números por letra de acordo com a tabela a seguir, além disso, deve remover os espaços em branco no começo e fim da string, se existirem.

**Tabela:**

```
1 : i  
3 : e  
4 : a  
5 : s  
0 : o
```

**Exemplo entrada:**

```
[' h3ll0 w0rld', ' w3b d3v3l0p3r ', '0tterw1s3', 'j4v4scr1pt ']
```

**Exemplo Saída:**

```
['helloworld', 'webdeveloper', 'otterwise', 'javascript']
```

**12 - [REFATORAÇÃO]** Dado o objeto abaixo, faça um programa que imprime no console o nome, cpf e telefone formatados. Dica: Separe o código em funções de formatação para cada um dos campos.

**Exemplo de Entrada 1:**

```
{nome: 'Analu dos santos baptista', cpf: '14776213931', telefone: '8328008258'}
```

**Exemplo de Saída 1:**

Analú dos Santos Baptista  
147.762.139-31  
(83) 2800-8258

### **Exemplo de Entrada 2:**

```
{nome: 'Cláudio da silva figueiredo ', cpf: '47187999887', telefone: '11988754092'}
```

### **Exemplo de Saída 2:**

Cláudio da silva figueiredo  
471.879.998-87  
(11) 98875-4092

**13 - [REFATORAÇÃO]** Em uma empresa de tecnologia a área de Recursos Humanos precisava organizar algumas informações a respeito dos funcionários. Para que isso fosse possível, ele precisava criar um programa que entregasse a lista de funcionalidades abaixo.

Tendo como entrada um array de objetos crie funções que realizam as seguintes operações:

- Ordena o Array em ordem crescente por idade;
- Ordena o Array em ordem decrescente por idade;
- Ordena o Array em ordem crescente por salário;
- Ordena o Array em ordem decrescente por salário;
- Ordena o Array em ordem crescente por senioridade;
- Ordena o Array em ordem decrescente por senioridade;

### **Exemplo de Entrada:**

```
[  
  { nome: "João", idade: 30, salario: 15000, senioridade: "senior" },  
  { nome: "Pedro", idade: 22, salario: 10000, senioridade: "pleno" },  
  { nome: "Carla", idade: 27, salario: 1500, senioridade: "estagio" },  
  { nome: "Lucas", idade: 35, salario: 15000, senioridade: "senior" },  
  { nome: "Roberta", idade: 45, salario: 5000, senioridade: "junior" },  
  { nome: "Patrícia", idade: 19, salario: 10000, senioridade: "pleno" },  
  { nome: "Joana", idade: 23, salario: 5000, senioridade: "junior" },  
]
```

**14** - Uma empresa gostaria de organizar seu quadro de funcionários, para isso contratou o estagiário Juquinha para criar um software. O CEO da empresa fez uma lista de pedidos para o Juquinha.

```
const employees = [  
  { id: 1, name: 'Carlos', age: 35, active: false, birth_date: '13/05/1986',  
    contract_date: '20/07/2015', job: 'front-end', type: 'clt' },  
  { id: 2, name: 'Matheus', age: 30, active: true, birth_date: '03/07/1991',  
    contract_date: '01/03/2020', job: 'front-end', type: 'clt'},  
  { id: 3, name: 'Pamela', age: 23, active: true, birth_date: '20/09/1998',  
    contract_date: '19/06/2021', job: 'back-end', type: 'pj'},  
  { id: 4, name: 'Fabiana', age: 32, active: false, birth_date: '08/12/1989',  
    contract_date: '13/04/2018', job: 'front-end', type: 'pj'},  
  { id: 5, name: 'João', age: 37, active: true, birth_date: '08/12/1984', contract_date:  
'03/08/2014', job: 'front-end', type: 'clt'},  
  { id: 6, name: 'Miguel', age: 29, active: true, birth_date: '08/12/1992',  
    contract_date: '26/02/2015', job: 'fullstack', type: 'clt'},  
  { id: 7, name: 'Francine', age: 27, active: false, birth_date: '27/01/1994',  
    contract_date: '23/05/2019', job: 'designer', type: 'pj'},  
  { id: 8, name: 'Matheus', age: 31, active: true, birth_date: '10/11/1990',  
    contract_date: '18/09/2017', job: 'marketing', type: 'clt'},  
  { id: 9, name: 'Gabriel', age: 28, active: true, birth_date: '08/12/1993',  
    contract_date: '14/11/2020', job: 'financeiro', type: 'clt'}  
]
```

Pedidos:

- A - Função para retornar apenas o funcionário com o id informado:
- B - Função para filtrar apenas os funcionários ativos na empresa:
- C - Função para filtrar apenas funcionários clt
- D - Função para criar uma nova estrutura onde cada funcionário receberá uma nova informação de base salarial
- E - Função onde separa em 2 grupos os funcionários pj e clt
- F - Função que retorna apenas funcionários ativos ordenados por ordem de idade

**15 - [REFATORAÇÃO]** O dono de uma livraria resolveu que precisava de novas funcionalidades no seu sistema de livros e para isso ele contratou um programador. Conversando com o programador e entendendo as necessidades que o dono da livraria precisava, eles chegaram nas seguintes funcionalidades:

- Autor do livro mais vendido;
- Filtrar livros que tem um preço abaixo do valor que será inserido;
- Filtrar livros que tem um preço acima do valor que será inserido;
- Pesquisa pelo nome do livro;
- Pesquisa pelo autor do livro;
- Pesquisa pelo código do livro;
- Lista do Estoque;
- Top X livros mais vendidos (X será um valor inserido como entrada);
- Aumento de X % no preço dos livros (X será um valor inserido como entrada).

Tendo por base as funcionalidades acima e como entrada um array de objetos, faça um programa que resolva o problema do dono da livraria.

### Exemplo de Entrada

```
[
  { cod: "8568014003",
    nome: "mais esperto que o diabo",
    autor: "Napoleon Hill",
    valor: 24.6,
    quantidadeVendido: 17675,},
  {cod: "8550801488",
    nome: "pai rico, pai pobre",
    autor: "Robert T. Kiyosaki",
    valor: 42.71,
    quantidadeVendido: 8992,},
  { cod: "8547001085",
    nome: "antifragil",
    autor: "Nassim Nicholas Taleb",
    valor: 70.99,
    quantidadeVendido: 1700,},
  {cod: "8595080801",
    nome: "o investidor inteligente",
    autor: "Benjamin Graham",
    valor: 36.3,
    quantidadeVendido: 8445, },
  { cod: "8539004119",
    nome: "o poder do hábito",
    autor: "Charles Duhigg",
```

```
    valor: 48.9,
    quantidadeVendido: 14581, },
{ cod: "8543102146",
  nome: "essencialismo",
  autor: "Greg Mckeown",
  valor: 36.43,
  quantidadeVendido: 9730, },
{ cod: "9788539003839",
  nome: "rápido e devagar",
  autor: "Daniel Kahneman",
  valor: 50.93,
  quantidadeVendido: 5703, },
{ cod: "8551003429",
  nome: "princípios",
  autor: "Ray Dalio",
  valor: 89.9,
  quantidadeVendido: 3707, },
{ cod: "855717358X",
  nome: "gestão de alta performance",
  autor: "Andrew S. Grove",
  valor: 34.94,
  quantidadeVendido: 442, },
{ cod: "8550805246",
  nome: "empresas feitas para vencer",
  autor: "Jim Collins",
  valor: 44.2,
  quantidadeVendido: 824, },
{ cod: "855080455X",
  nome: "avalie o que importa",
  autor: "John Doerr",
  valor: 40.9,
  quantidadeVendido: 797, },
];
```