

OBS: Você deve utilizar os conceitos vistos na última aula para realizar os exercícios abaixo. Procure não buscar a resposta nos exercícios que você já resolveu, resolva o problema como se fosse a primeira vez que você está se deparando com ele.

1 - [REFATORAÇÃO] Pegue um valor de entrada e calcule o menor número de notas possíveis (cédulas) no qual o valor pode ser decomposto. As notas consideradas são de 100, 50, 20, 10, 5, 2 e 1. A seguir mostre o valor lido e a relação de notas necessárias.

Exemplo de Entrada	Exemplo de Saída
576	576 5 nota(s) de R\$ 100,00 1 nota(s) de R\$ 50,00 1 nota(s) de R\$ 20,00 0 nota(s) de R\$ 10,00 1 nota(s) de R\$ 5,00 0 nota(s) de R\$ 2,00 1 nota(s) de R\$ 1,00

2 - [REFATORAÇÃO] Tendo como informação dois valores, valor1 e valor2, faça um programa que imprima os dois valores no console. Após imprimir, troque a informação de valor1 pela informação do valor2 e imprima no console novamente.

Exemplo de Entrada 1:

3
12

Exemplo de Entrada 2:

25
23

Exemplo de Saída 1:

Valor 1 = 3
Valor 2 = 12
Valor 1 = 12
Valor 2 = 3

Exemplo de Saída 2:

Valor 1 = 25
Valor 2 = 23
Valor 1 = 23
Valor 2 = 25

3 - [REFATORAÇÃO] Uma pessoa desenvolvedora recebe frequentemente propostas de trabalho pela CLT(Consolidação das Leis Trabalhistas) e pela PJ (Pessoa Jurídica). Considerando que o trabalho como CLT envolve férias, 13º salário, R\$ 700 de benefício e 8% mensal referente ao FGTS. Faça um programa que receba como entrada os dois valores mensais oferecidos e tome a decisão informando qual proposta a pessoa deve aceitar.

4 - [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

5 - [REFATORAÇÃO] Crie um algoritmo que tem como entrada um número e imprime a sua tabuada do 1 ao 10.

OBS: Para melhorar a resolução desse problema, monte um código que aceite um segundo valor e irá imprimir a tabuada de 1 até esse valor.

6 - [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 imprimem 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]

7 - [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]

8 - [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]

9 - [REFATORAÇÃO] Dado um array de números, faça um programa que encontre o menor elemento deste array e sua posição. Mostre as informações encontradas no console.

OBS: Para melhorar a resolução desse problema, monte um programa que também consegue retornar qual o maior valor e qual a sua posição.

Exemplo de Entrada:

[1,2,3,4,-5,6,7,8,9,10]

Exemplo de Saída:

Menor valor: -5

Posição: 4

10 - [REFATORAÇÃO] Dado a lista de compras a seguir, faça um programa que imprime no console o 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

11 - [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;

OBS: Tente melhorar esse exercício utilizando os conhecimentos novos que você aprendeu na aula anterior.

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" },
]
```

12 - [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, },
];
```

13 - Faça um programa que recebe um array e uma função como parâmetro e percorre todo o array aplicando a função em cada um dos elementos do array.

Exemplo de Entrada:

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

multiplicaPorDez

Exemplo de Entrada 2:

[10, 80, 60, 130, 70, 90, 150, 220, 3560, 40, 30]

dividePorCinco

Exemplo de Saída:

[10, 80, 60, 130, 70, 90, 150, 220, 3560, 40, 30]

Exemplo de Saída 2:

[2, 16, 12, 26, 14, 18, 30, 44, 712, 8, 6]

14 - Uma loja online que possui a estrutura do objeto abaixo, precisa que algumas funcionalidades sejam implementadas. Implemente a lista de funcionalidades abaixo.

Estrutura do Objeto:

```
{ cod:
  name:
  stock:
  price:
  category:
}
```

Lista de Funcionalidades:

- Filtro por nome, código e categoria;
- Filtro por faixa de preço;
- Ordenação em Ordem Crescente e Decrescente por todos os campos;

Transação de Compra:

Além das funcionalidades acima, o usuário poderá realizar a compra.

- O processo de compra será uma função que confere se o produto existe e se ele possui estoque, a quantidade será passada como parâmetro junto com o código do produto;

- Um cliente pode realizar várias compras, então todos os produtos selecionados devem ser adicionados a um “carrinho de compras”;
- Deve existir uma função que mostra o valor total que existe atualmente no carrinho;
- O programa deve possuir uma função de finalização da compra em que o programa recebe como parâmetro o valor total e a forma de pagamento:
 - Boletos tem 10% de desconto no valor total;
 - Cartão de Débito tem 15% de desconto no valor total;
 - A opção crédito, deve receber quantas parcelas o cliente deseja parcelar (máximo de 12x), em cada parcela deve incidir 3% de juros, sendo assim o cliente deve saber o valor da parcela e o total.
- O programa deve mostrar ao final um relatório em que são mostradas as compras e a forma de pagamento com o valor total (e o número de parcelas caso a opção escolhida tenha sido crédito).