

Info.: Os programas das questões abaixo devem ser codificados na linguagem Javascript

Git HUB - Click aqui para acessar o código no GitHub

1. Implemente um algoritmo que leia as notas de 10 alunos armazenando-as em um vetor de 10 posições. Ao final escreva na tela somente as notas maiores que 5.0.

//src/unidade_III/lista01/questao01.js

```
const reader = require("readline-sync");

const notas = [];

for (let i = 0; i < 10; i++) {
    notas.push(reader.questionFloat(`Informe a nota do aluno ${i + 1}: `));
}

console.log(`Notas maiores que 5.0: ${notas.filter(nota => nota > 5.0).join(", ")}`);
```

2. Implemente um algoritmo que leia as notas e os nomes de 5 alunos armazenando os dados em vetores de 5 posições, sendo que as notas serão armazenadas em um vetor de reais e os nomes em um outro vetor do tipo cadeia de caracteres. Ao final o algoritmo deve escrever na tela somente os nomes dos alunos que tiraram nota maior que 5.0.

//src/unidade_III/lista01/questao02.js

```
const reader = require("readline-sync");

const notas = [];
const nome = [];

for (let i = 0; i < 5; i++) {
    nome.push(reader.question(`Informe o nome do aluno ${i + 1}: `));
    notas.push(reader.questionFloat(`Informe a nota do aluno ${i + 1}: `));
}

for (let i = 0; i < 5; i++) {
    if (notas[i] > 5.0) {
        console.log(`Aluno: ${nome[i]} - Nota: ${notas[i]}`);
    }
}
```

3. Elabore um algoritmo que armazene o nome e duas notas de 5 alunos e imprima uma listagem contendo nome, as duas notas e a média de cada aluno.

//src/unidade_III/lista01/questao03.js

```
const reader = require("readline-sync");

const nome = [];

const notas1 = [];
const notas2 = [];

for (let i = 0; i < 5; i++) {
    nome.push(reader.question(`Informe o nome do aluno ${i + 1}: `));
    notas1.push(reader.questionFloat(`Informe a nota 1 do aluno ${i + 1}: `));
    notas2.push(reader.questionFloat(`Informe a nota 2 do aluno ${i + 1}: `));
}

for (let i = 0; i < 5; i++) {
    console.log(`Aluno: ${nome[i]} - Nota 1: ${notas1[i]} - Nota 2: ${notas2[i]} - Média: ${(notas1[i] + notas2[i]) / 2}`);
}
```

4. Elabore um algoritmo que leia a nota de 80 alunos e que imprima ao final a nota de cada aluno e a média da turma

```
//src/unidade_III/lista01/questao04.js
```

```
const reader = require("readline-sync");
```

```
// Elabore um algoritmo que leia a nota de 80 alunos e que imprima ao final a nota de cada aluno e a média da turma
```

```
const notas = [];
```

```
for (let i = 0; i < 80; i++) {  
    notas.push(reader.questionFloat(`Informe a nota do aluno ${i + 1}: `));  
}
```

```
console.log(`Notas: ${notas}`);
```

```
console.log(`Média da turma: ${notas.reduce((acc, nota) => acc + nota) / notas.length}`);
```

5. Crie um script que pergunte 3 números ao usuário, coloque em um array, depois exiba tal array. Modifique os elementos do array de modo a sequência de números ficar do contrário. Ou seja, se digitou: 1,2,3 Vai aparecer: 3,2,1

```
//src/unidade_III/lista01/questao05.js
```

```
const reader = require("readline-sync");
```

```
const numeros = [];
```

```
for (let i = 0; i < 3; i++) {  
    numeros.push(reader.questionInt(`Informe o número ${i + 1}: `));  
}
```

```
console.log(`Números: ${numeros}`);
```

```
console.log(`Números ao contrário: ${numeros.reverse()}`);
```

6. Refaça o script anterior, mas agora pergunte ao usuário quantos números ele deseja inserir no array. Depois, o usuário vai preenchendo elemento por elemento do vetor. Ao final, exiba o array e o array ao contrário (generalização do script anterior).

```
//src/unidade_III/lista01/questao06.js
```

```
const reader = require("readline-sync");
```

```
const numeros = [];
```

```
tamanho = reader.questionInt("Informe o tamanho do array: ");
```

```
for (let i = 0; i < tamanho; i++) {  
    numeros.push(reader.questionInt(`Informe o número ${i + 1}: `));  
}
```

```
console.log(`Números: ${numeros}`);
```

```
console.log(`Números ao contrário: ${numeros.reverse()}`);
```

7. Crie uma função que recebe um vetor e um número. Ela deve mostrar todos os índices onde esse número aparece no vetor, e não só apenas o primeiro e/ou último índice em que o número aparece.

```
//src/unidade_III/lista01/questao07.js
```

```
function mostrarIndices(vetor, numero) {  
  const indices = [];  
  for (let i = 0; i < vetor.length; i++) {  
    if (vetor[i] === numero) {  
      indices.push(i);  
    }  
  }  
  return indices;  
}  
  
indices = mostrarIndices([1, 5, 3, 4, 5, 6, 7, 5, 9, 10], 5);  
  
console.log(`Índices: ${indices}`);
```

8. Dada uma coleção de N números, imprimir o índice do primeiro número negativo, se houver.

```
//src/unidade_III/lista01/questao08.js
```

```
const reader = require("readline-sync");  
  
// Dada uma coleção de N números, imprimir o índice do primeiro número negativo, se houver.  
  
const numeros = [];  
  
tamanho = reader.questionInt("Informe o tamanho do array: ");  
  
for (let i = 0; i < tamanho; i++) {  
  numeros.push(reader.questionInt(`Informe o número ${i + 1}: `));  
}  
  
for (let i = 0; i < numeros.length; i++) {  
  if (numeros[i] < 0) {  
    console.log(`O índice do primeiro número negativo é ${i}`);  
    break;  
  }  
}
```