

ATIVIDADE PRÁTICA

EXERCÍCIOS COMPLEMENTARES

Instruções gerais:

Esta Lista de Exercícios é Opcional e não será entregue no Canvas

Boas práticas:

1. Esta lista de exercícios foi elaborada para ajudá-lo a praticar **Lógica de Programação** e os **Conceitos Fundamentais** da Linguagem de Programação que você está estudando.
2. A lista contém **20 exercícios**, organizados nas seguintes categorias:
 - **Variáveis e Operadores**
 - **Laços Condicionais**
 - **Laços de Repetição**
 - **Vetores**
 - **Matrizes**
3. Leia **atentamente o enunciado** de cada exercício antes de iniciar a resolução.
4. Observe as **indicações de Entrada e Saída** esperadas para cada exercício.
5. Preste atenção nos **desenhos, diagramas, fórmulas e dicas** fornecidas, pois elas o ajudarão a simplificar a solução.
6. Utilize o **Cookbook**, os **Vídeos da Plataforma** e os **Códigos guia** como referências ao longo da resolução.
7. Se restar alguma dúvida, entre em contato com os **instrutores da sua turma pelo Discord**.
8. Após finalizar a implementação de todos os exercícios, **envie os códigos para um repositório no seu GitHub**.

VARIÁVEIS E OPERADORES

- 1) Elabore um algoritmo para calcular a área de um círculo. O algoritmo deverá solicitar ao usuário, via teclado, a entrada de um número do tipo *float*, que representará o raio do círculo. Em seguida, o programa **calculará e exibirá a área do círculo**. Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite o raio do círculo: 2.00	A área do círculo é: 12.57
Digite o raio do círculo: 100.64	A área do círculo é: 31819.34

Fórmula para o cálculo da área do círculo:

$$area = \pi * r^2$$

$$\pi = 3.1415$$

→ **Dica:** Geralmente as Linguagens de Programação possuem Bibliotecas nativas para cálculos matemáticos, que fornecem a constante *PI* e a *Potenciação*.

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados

2) Elabore um algoritmo para calcular e exibir a quantidade de litros de combustível consumidos durante uma viagem com um automóvel que possui rendimento de **12 km/L**. O cálculo do consumo de combustível deve ser feito com base em dois parâmetros que serão lidos via teclado:

- **Tempo gasto na viagem** (em horas);
- **Velocidade média** durante a viagem (em km/h).

Com essas informações, é possível calcular a **distância percorrida** e, em seguida, determinar a **quantidade de combustível necessária**. O resultado deve ser apresentado com **duas casas decimais** após o ponto. Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite o tempo gasto na viagem (em horas): 10	Total de combustível gasto (em litros): 70.83
Digite a velocidade média (em KM por horas): 85	

Fórmula para o cálculo da distância:

$$distancia = tempoViagem * velocidadeMedia$$

Fórmula para o cálculo do consumo do combustível:

$$consumoCombustivel = distancia / rendimento$$

- 3) Elabore um algoritmo que leia via teclado um número inteiro que representa um intervalo de tempo em segundos. Na sequência exiba este tempo na tela expresso em **horas:minutos:segundos**. Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite o tempo: 556	Tempo expresso em (hh:mm:ss): 0:9:16
Digite o tempo: 140153	Tempo expresso em (hh:mm:ss): 38:55:53

Fórmulas:

- **Calcular as Horas:** $\lfloor tempo / 3600 \rfloor$
- **Calcular os Minutos:** $\lfloor (tempo \% 3600) / 60 \rfloor$
- **Calcular os Segundos:** $tempo \% 60$

→ **Dica:** O símbolo $\lfloor \rfloor$ inserido nas fórmulas para o cálculo das horas e dos minutos, é a função **floor (piso)**, que tem o objetivo de obter apenas a parte inteira da divisão. A função floor geralmente está disponível na Biblioteca nativa para cálculos matemáticos da linguagem de programação.

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados

- 4) Elabore um algoritmo que leia via teclado um número inteiro que representa um valor em Reais (R\$). Na sequência exiba **o menor número de notas (Cédulas de dinheiro), do qual este valor pode ser decomposto**. Para o exercício, considere as notas de R\$ 200.00, R\$ 100.00, R\$ 50.00, R\$ 20.00, R\$ 10.00, R\$ 5.00, R\$ 2.00 e R\$ 1.00. Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite um valor em Reais (sem os centavos): 576.00	<p>Para o valor R\$ 576,00 utilizaremos:</p> <p>2 nota(s) de R\$ 200,00 1 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</p>

Fórmulas:

- **Definir o Valor Restante Inicial:** $valorRestante = valor$
- **Calcular o número de Notas:** $\lfloor valorRestante / valorNota \rfloor$
- **Atualizar o Valor Restante:** $valorRestante \% = valorNota$

- **Dica 01:** A variável **valorRestante** armazena o valor que sobra após a decomposição em cada tipo de nota. Por exemplo, para um valor de R\$ 250,00, é utilizada uma nota de R\$ 200,00, restando R\$ 50,00.
- **Dica 02:** O símbolo $\lfloor \rfloor$, presente nas fórmulas para o cálculo das horas e dos minutos, representa a função **floor** (piso). Essa função retorna apenas a parte inteira da divisão e está disponível na biblioteca nativa de cálculos matemáticos da maioria das linguagens de programação.
- **Dica 03:** A fórmula para calcular as notas e atualizar o valor restante deve ser aplicada sequencialmente para todas as denominações, começando pela nota de R\$ 200,00 e seguindo até a de R\$ 1,00.

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados

LAÇOS CONDICIONAIS

- 5) Elabore um algoritmo que leia 3 valores inteiros e ordene-os tanto em ordem crescente quanto em ordem decrescente. No final, o programa deve exibir os valores na seguinte sequência:
- Na sequência original;
 - Em ordem crescente;
 - Em ordem decrescente.

Cada sequência deve ser exibida em uma linha separada, com uma linha em branco entre elas. Veja o exemplo abaixo:

ENTRADA	SAÍDA
Digite o primeiro número inteiro: 7	Sequência original: 7 21 -14
Digite o segundo número inteiro: 21	Ordem crescente: -14 7 21
Digite o terceiro número inteiro: -14	Ordem decrescente: 21 7 -14

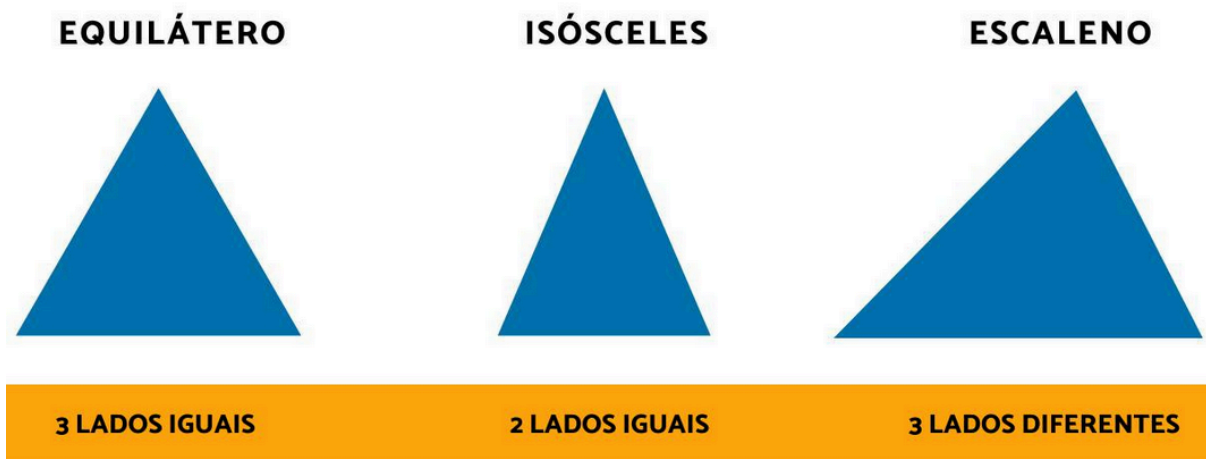
Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais

6) Leia três valores de ponto flutuante A, B e C, que representam os lados de um triângulo. Em seguida, determine o tipo de triângulo que esses três lados formam, com base nas condições abaixo, exibindo a mensagem correspondente:

- Se $A \geq B + C$, o programa deve exibir: **NÃO FORMA TRIÂNGULO**.
- Se todos os lados forem iguais, o programa deve exibir: **TRIÂNGULO EQUILÁTERO**.
- Se apenas dois lados forem iguais, o programa deve exibir: **TRIÂNGULO ISÓSCELES**.
- Se todos os lados forem diferentes e formarem um triângulo, o programa deve exibir: **TRIÂNGULO ESCALENO**.

Veja a imagem abaixo:



Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite o tamanho do lado A: 6.0 Digite o tamanho do lado B: 6.0 Digite o tamanho do lado C: 6.0	TRIANGULO EQUILÁTERO
Digite o tamanho do lado A: 5.0 Digite o tamanho do lado B: 7.0 Digite o tamanho do lado C: 9.0	TRIANGULO ESCALENO
Digite o tamanho do lado A: 6.0 Digite o tamanho do lado B: 6.0 Digite o tamanho do lado C: 10.0	TRIANGULO ISÓSCELES
Digite o tamanho do lado A: 10.0 Digite o tamanho do lado B: 20.0 Digite o tamanho do lado C: 30.0	NÃO FORMA TRIANGULO

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais

- 7) Elabore um algoritmo que leia via teclado um número inteiro que representa o DDD de uma cidade. Na sequência exiba **o nome da cidade o qual o DDD pertence**. Para o exercício, considere a tabela de cidades abaixo. Caso o DDD não seja encontrado, exiba a mensagem **DDD não encontrado!**

DDD	CIDADE
11	São Paulo
21	Rio de Janeiro
27	Vitória
31	Belo Horizonte

Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite um DDD: 11	O DDD 11 pertence à cidade de São Paulo
Digite um DDD:	DDD não encontrado!

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais

8) A empresa Portugal TI resolveu conceder um aumento de salários a seus funcionários conforme a tabela abaixo:

SALÁRIO	PERCENTUAL DE REAJUSTE
R\$ 1412,00	15%
R\$ 1412,01 a R\$ 2824,00	12%
R\$ 2824,01 a R\$ 4236,00	10%
R\$ 4236,01 a R\$ 5648,00	7%
Acima de R\$ 5648,00	5%

Elabora um algoritmo que leia via teclado o salário do funcionário (numero float) e calcule e mostre o **novo salário**, bem como o **valor de reajuste** ganho e o **índice percentual do reajuste**. Veja o exemplo abaixo:

ENTRADA	SAÍDA
Digite o valor do salário (R\$): 3000.00	<p>Novo salario: R\$ 3300.00</p> <p>Valor do reajuste: R\$ 300.00</p> <p>Percentual do aumento: 10%</p>

Fórmula para o cálculo do salário:

$$reajuste = (salarioAtual * percentual) / 100$$

$$novoSalario = salarioAtual + reajuste$$

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais

9) A fabricação de presentes para o Natal é um processo desafiador. Muitas vezes, os duendes trabalham até tarde para garantir que tudo esteja pronto a tempo e com perfeição. Para organizar melhor suas tarefas, os duendes calcularam o tempo necessário, em minutos, para fabricar cada presente.

Agora, com o expediente chegando ao fim, o duende Ed precisa de sua ajuda! Ele ainda tem dois presentes para fabricar e faltam N minutos para o término do expediente. Sua missão é ajudá-lo a descobrir se ele conseguirá finalizar os dois presentes hoje ou se precisará adiar o trabalho para amanhã.

Elabore um algoritmo que:

- Leia **um número inteiro N (entre 0 e 100)**, indicando os minutos restantes no expediente.
- Leia **dois números inteiros A e B (entre 0 e 100)**, representando o tempo necessário para fabricar cada um dos dois presentes.

Calcule se o tempo restante é suficiente para concluir ambos os presentes e exiba:

- **"Farei hoje!"** se for possível fabricar os dois presentes dentro do tempo disponível.
- **"Deixa para amanhã!"** caso contrário.

Veja os exemplos abaixo:

ENTRADA	SAÍDA
Tempo restante (min): 12 Tempo de fabricação do presente 01 (min): 5 Tempo de fabricação do presente 02 (min): 4	Farei hoje!
Tempo restante (min): 15 Tempo de fabricação do presente 01 (min): 10 Tempo de fabricação do presente 02 (min): 8	Deixa para amanhã!

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais

LAÇOS DE REPETIÇÕES

- 10) Desenvolva um algoritmo para determinar todos os divisores de um número inteiro. O algoritmo deverá solicitar ao usuário a entrada via teclado de um número inteiro e, em seguida, calcular e exibir todos os divisores inteiros que o dividem exatamente (resto da divisão igual a zero). Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite um número inteiro: 6	Os divisores positivos do número 6 são: 1 2 3 6
Digite um número inteiro: 30	Os divisores positivos do número 30 são: 1 2 3 5 6 10 15 30

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição

11) Elabore um algoritmo que leia 2 valores inteiros **X** e **Y**. A seguir, calcule e mostre a **soma de todos os números ímpares entre eles**. Veja o exemplo abaixo:

ENTRADA	SAÍDA
Digite o primeiro número inteiro: 2 Digite o segundo número inteiro: 10	A Soma de todos os números ímpares é 24

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição

12) Elabore um algoritmo que leia 1 valor inteiro **N**, entre 5 e 2000 e a seguir, calcule e mostre **todos os números pares entre 1 e N (incluindo N), elevados ao quadrado**. Veja o exemplo abaixo:

ENTRADA	SAÍDA
Digite o primeiro número inteiro: 6	$2^2 = 4$ $4^2 = 16$ $6^2 = 36$

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição

- 13) Elabore um algoritmo que leia 1 valor inteiro **N**, entre 1 e 1000 e a seguir, calcule e mostre **todos os números entre 1 e N (incluindo N), elevados ao quadrado e ao cubo**. Veja o exemplo abaixo:

ENTRADA	SAÍDA
Digite o primeiro número inteiro: 5	1 1 1 2 4 8 3 9 27 4 16 64 5 25 125

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição

- 14) O Natal traz tanta alegria que você sente vontade de gritar para o mundo: "**Feliz Nataaaaal!!**". Para expressar toda essa felicidade, crie um algoritmo que receba um número inteiro **I** (entre 1 e 100), representando o seu **índice de felicidade**. Com base nesse índice, determine quantas vezes a letra '**a**' será repetida na última palavra da frase "*Feliz Natal*". Veja o exemplo abaixo:

ENTRADA	SAÍDA
Digite o primeiro número inteiro: 5	Feliz Nataaaaal!

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição

VETORES

- 15)Elabore um algoritmo que leia um vetor de inteiros com 10 posições. No final, mostre todas as posições do vetor que armazenam um valor menor ou igual a 10 e o valor armazenado em cada uma das posições.

ENTRADA	SAÍDA
Digite o primeiro número inteiro: 5 Digite o segundo número inteiro: 10 Digite o terceiro número inteiro: 7 Digite o quarto número inteiro: 15 Digite o quinto número inteiro: 20 Digite o sexto número inteiro: 25 Digite o sétimo número inteiro: 2 Digite o oitavo número inteiro: 1 Digite o nono número inteiro: 3 Digite o décimo número inteiro: 35	Números menores do que 10: 5 7 2 1 3

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- Vetores

16)Elabore um algoritmo que crie um vetor de inteiros com 20 posições e exiba os dados do vetor na tela. Na sequência, troque o primeiro elemento com o último, o segundo elemento com o penúltimo, o terceiro com o antepenúltimo e assim sucessivamente, até trocar o 10º com o 11º valor. No final, mostre o vetor modificado.

ENTRADA	SAÍDA
vetor = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]	Vetor original: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] Vetor modificado: [20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- Vetores

- 17)Elabore um algoritmo que crie um vetor de inteiros com 100 posições e leia um número **T** inteiro entre 2 e 50. Preencha o vetor de inteiros com uma sequência repetitiva de números de **0 até T - 1**. O ciclo deve se repetir até que todas as 100 posições do vetor sejam preenchidas. Após preencher o vetor, exiba o conteúdo na tela. Veja o exemplo abaixo:

ENTRADA	SAÍDA
Digite um número inteiro entre 2 e 50: 5	N[0] = 0 N[1] = 1 N[2] = 2 N[3] = 3 N[4] = 4 N[5] = 0 N[6] = 1 N[7] = 2 N[8] = 3 N[9] = 4 ... N[95] = 0 N[96] = 1 N[97] = 2 N[98] = 3 N[99] = 4

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- Vetores

MATRIZES

18) Escreva um algoritmo que leia um número inteiro **N** entre 0 e 100, representando a ordem de uma matriz quadrada **M**, e construa a matriz seguindo o padrão descrito abaixo:

Para cada valor **N** lido (exceto o 0), imprima a matriz correspondente com as seguintes regras de formatação:

- Cada elemento da matriz deve ocupar um campo de tamanho **3**, alinhado à direita e separado por um espaço.
- Não devem existir espaços extras ao final de cada linha da matriz.
- Após a exibição de cada matriz, insira uma linha em branco.

Veja os exemplos abaixo:

ENTRADA	SAÍDA
Digite a ordem da matriz: 3	1 1 1 1 2 1 1 1 1
Digite a ordem da matriz: 5	1 1 1 1 1 1 2 2 2 1 1 2 3 2 1 1 2 2 2 1 1 1 1 1 1

- **Dica 01:** Para eliminar os espaços em branco no início e no final da string, utilize a função **trim()**.
- **Dica 02:** Para adicionar os 3 espaços em branco no final do valor, converta o valor para string e utilize a função **padStart()**.

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- Matrizes

19) Crie um algoritmo que leia os seguintes valores via teclado:

- Um número inteiro **N** (entre 0 e 11), representando uma linha específica de uma matriz onde será realizada uma determinada operação matemática.
- Um caractere maiúsculo **O** ('S' ou 'M'), indicando a operação matemática que será executada:
 - **'S'**: Soma dos elementos.
 - **'M'**: Média dos elementos.

Em seguida, faça o seguinte:

- Crie uma matriz **M[12][12]**, preenchida com números inteiros aleatórios.
- Identifique os elementos da **área verde** da matriz (correspondente à linha **N**) e, conforme a operação indicada ('S' ou 'M'), calcule e exiba o resultado.

A figura abaixo ilustra um exemplo em que o valor **N = 2** é informado, destacando os elementos da área verde que serão considerados na operação.

	0	1	2	3	4	5	6	7	8	9	10	11
0	1	3	5	8	2	1	2	6	3	4	6	5
1	10	2	7	9	5	4	8	2	1	6	5	1
2	5	7	8	10	2	3	7	9	6	2	5	3
3	3	0	1	2	5	4	7	9	6	5	2	0
4	9	8	7	4	5	6	3	2	1	0	0	1
5	5	2	0	3	6	4	8	5	2	9	2	1
6	1	3	5	8	2	1	2	6	3	4	6	5
7	10	2	7	9	5	4	8	2	1	6	5	1
8	3	0	1	2	5	4	7	9	6	5	2	0
9	9	8	7	4	5	6	3	2	1	0	0	1
10	5	2	0	3	6	4	8	5	2	9	2	1
11	1	3	5	8	2	1	2	6	3	4	6	5

Veja o exemplo abaixo, baseado na matriz acima:

ENTRADA	SAÍDA
<p>Digite um número inteiro entre 1 e 11: 2</p> <p>Digite a operação (S ou M): S</p>	<p>A soma de todos os elementos da linha 2 é 67</p>
<p>Digite um número inteiro entre 1 e 11: 2</p> <p>Digite a operação (S ou M): M</p>	<p>A média de todos os elementos da linha 2 é 5,58</p>

→ **Dica:** Para gerar números aleatórios inteiros na matriz, utilize a combinação das funções **floor()** e **random()**. A função **random** gera números aleatórios.

Exemplo - gerar números aleatórios entre 1 e 10:

```
Math.floor(Math.random() * 10) + 1
```

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- Matrizes

20) Crie um algoritmo que leia os seguintes valores via teclado:

- Um número inteiro **N** (entre 0 e 11), representando uma coluna específica de uma matriz onde será realizada uma determinada operação matemática.
- Um caractere maiúsculo **O** ('S' ou 'M'), indicando a operação matemática que será executada:
 - **'S'**: Soma dos elementos.
 - **'M'**: Média dos elementos.

Em seguida, faça o seguinte:

- Crie uma matriz **M[12][12]**, preenchida com números inteiros aleatórios.
- Identifique os elementos da **área verde** da matriz (correspondente à coluna **N**) e, conforme a operação indicada ('S' ou 'M'), calcule e exiba o resultado.

A figura abaixo ilustra um exemplo em que o valor **N = 2** é informado, destacando os elementos da área verde que serão considerados na operação.

	0	1	2	3	4	5	6	7	8	9	10	11
0	1	3	5	8	2	1	2	6	3	4	6	5
1	10	2	7	9	5	4	8	2	1	6	5	1
2	5	7	8	10	2	3	7	9	6	2	5	3
3	3	0	1	2	5	4	7	9	6	5	2	0
4	9	8	7	4	5	6	3	2	1	0	0	1
5	5	2	0	3	6	4	8	5	2	9	2	1
6	1	3	5	8	2	1	2	6	3	4	6	5
7	10	2	7	9	5	4	8	2	1	6	5	1
8	3	0	1	2	5	4	7	9	6	5	2	0
9	9	8	7	4	5	6	3	2	1	0	0	1
10	5	2	0	3	6	4	8	5	2	9	2	1
11	1	3	5	8	2	1	2	6	3	4	6	5

Veja o exemplo abaixo, baseado na matriz acima:

ENTRADA	SAÍDA
<p>Digite um número inteiro entre 0 e 11: 2</p> <p>Digite a operação (S ou M): S</p>	<p>A soma de todos os elementos da coluna 2 é 54</p>
<p>Digite um número inteiro entre 0 e 11: 2</p> <p>Digite a operação (S ou M): M</p>	<p>A média de todos os elementos da coluna 2 é 4,50</p>

→ **Dica:** Para gerar números aleatórios inteiros na matriz, utilize a combinação das funções **floor()** e **random()**. A função **random** gera números aleatórios.

Exemplo - gerar números aleatórios entre 1 e 10:

```
Math.floor(Math.random() * 10) + 1
```

Na construção do Algoritmo, utilize os seguintes conteúdos:

- Variáveis
- Entrada e Saída de dados
- Laços Condicionais
- Laços de Repetição
- Matrizes