

# Desafios Repetição e Vetores

O mais fácil é o do Supermercado e o mais difícil é o Don't Blink

## Login

Desenvolva um programa que simule um sistema de validação de senhas. O programa deve conter uma lista de senhas fictícias e verificar se a senha fornecida pelo usuário corresponde a uma das senhas da lista.

Crie uma lista contendo o cadastro de 10 usuários fictícios. Os usuários deverão ter um email e uma senha, bem como informações adicionais como nome entre outras que desejar. As senhas devem ser diferentes entre si e não devem ser senhas reais para garantir a segurança.

Exemplo de tabela:

Nome	Email	Senha
Alice Silva	alice.silva@example.com	senha123
Bruno Costa	bruno.costa@example.com	123456
Carla Oliveira	carla.oliveira@example.com	abcdef
Daniel Souza	daniel.souza@example.com	qwerty
Elisa Mendes	elisa.mendes@example.com	password1
Fabio Lima	fabio.lima@example.com	123abc
Gabriela Rocha	gabriela.rocha@example.com	654321
Henrique Dias	henrique.dias@example.com	passw0rd
Isabel Teixeira	isabel.teixeira@example.com	senha321
João Almeida	joao.almeida@example.com	abc123

O seu programa deve receber do usuário via argumentos da linha de comando um e-mail e uma senha, então pesquise na tabela acima se existe alguma correspondência, se houver exiba alguma tela de bem vindo com as informações completas do usuário (Incluindo o nome por exemplo), porém se errar deve terminar o programa informando que o email ou a senha estão incorretos.

## Gabarito

Um professor de português aplicou uma prova de múltipla escolha para seus alunos, com 6 questões.

Nome	Questões Marcadas
marcio	bdeead
junior	acbcce
marcos	abbade
robert	abbade
ariana	abbade
meghan	babaca
juceli	adaadd
tiffany	abcecb
penina	aeddbc
denise	bebebe
adriel	cabeca

Após ver as questões marcadas, o professor ficou desconfiado que alguns alunos colaram e gostaria de saber se houve marcações iguais por vários alunos.

Faça um programa que, analisando a tabela acima, escreva um por linha a relação dos alunos que possivelmente colaram, isto é, aqueles que suas respostas são idênticas a pelo menos algum outro aluno, não importa qual ocorrência foi encontrada a correspondência das respostas, apenas escreva a lista dos nomes.

(O programa deve funcionar para qualquer lista que for provida, não necessariamente a exibida acima)

## IMC

O Índice de Massa Corporal (IMC) é uma medida internacional usada para calcular se uma pessoa está no peso ideal. O IMC é determinado pela divisão da massa do indivíduo pelo quadrado de sua altura, onde a massa está em quilogramas e a altura em metros.

A fórmula é:

$$\text{IMC} = \text{Massa} / (\text{Altura} * \text{Altura})$$

O resultado do IMC é então comparado com a tabela a seguir, que indica o grau de obesidade do indivíduo:

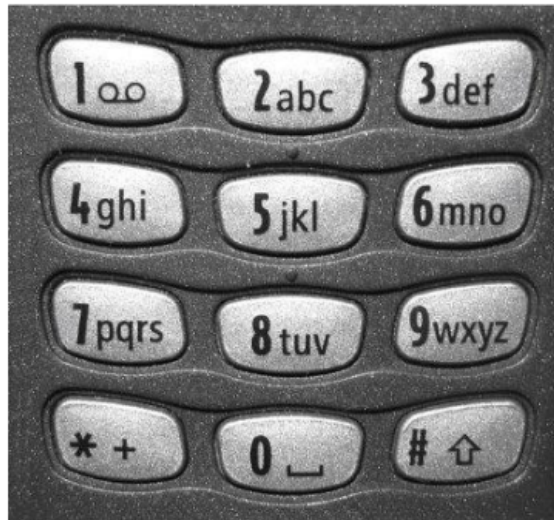
IMC	Classificação
menos que 16	Magreza grave
16 a 17	Magreza moderada
17 a 18,5	Magreza leve
18,5 a 25	Saudável
25 a 30	Sobrepeso
30 a 35	Obesidade Grau I
35 a 40	Obesidade Grau II (severa)
mais que 40	Obesidade Grau III (mórbida)

Faça um programa que receba a massa e a altura do usuário via argumentos da linha de comando e exiba a classificação do IMC dessa pessoa com base na tabela fornecida.

Obs: Deve fazer utilizando uma matriz para guardar as informações da tabela acima, então vá passando pela lista com imc's e caso encontre um valor que é maior que o imc calculado, exibe a última classificação.

## Teclado Numérico

Em celulares antigos, e muitos telefones, há letras em cada dígito.



Faça um programa que a partir de uma entrada de texto (via argumentos da linha de comando), diga qual sequência de números deve ser digitada em um teclado assim, para escrever essa palavra (Ignorando maiúsculas ou minúsculas).

Para simplificar, seria interessante ter uma tabela, onde na primeira coluna se tem as letras de todo o alfabeto, e na segunda coluna os dígitos (Exibindo apenas as primeiras letras, o resto se segue o padrão):

Letra	Dígito
A	2
B	22
C	222
D	3
E	33

### Exemplos de conversões:

ERICK → 3377744422255

WEIL → 93344455

TIJOLO → 84445666555666

(O programa deve receber o texto via argumentos da linha de comando e então escrever os dígitos)

## Sistema de Pesquisa de Suspeitos por Altura

Uma delegacia precisa de um sistema que permita pesquisar suspeitos com base na altura, a partir de uma base de dados fornecida. O programa deve ser escrito em JavaScript, utilizando Node.js, e deve receber a entrada via argumentos da linha de comando.

### Lista de Suspeitos

Nome	Altura (m)
Ivonildo	1.90
Juscelino	1.50
Gilmar	1.55
Severino	1.70
Melquisedeque	1.85
Leon	1.95
Uberlindo	1.60

Faça um programa que leia do usuário a altura que a vítima diz que o suspeito tem, e com uma margem de erro de 10 centímetros exiba quais são os possíveis suspeitos em torno da altura informada.

Exemplo de execução:

Altura Informada: 1.60

Suspeitos:

Juscelino: 1.50 m

Gilmar: 1.55 m

Uberlindo: 1.60 m

Severino: 1.70 m

## Supermercado

Crie um programa que receba como entrada a quantidade de reais disponíveis para um usuário, via argumentos da linha de comando. O programa deve analisar uma lista de produtos e seus preços, informando quais produtos o usuário pode comprar com a quantia disponível. A saída deve listar os produtos possíveis, um por linha.

Lista de Produtos	
Nome	Preço (R\$)
Arroz pacote 5kg	13.90
Contra-Filé 1kg	24.90
Frango	20.90
Feijão pacote 1kg	5.80
Trigo 1kg	2.10
Polvilho 1kg	3.10
Pão 4 un.	2.00
Leite 1L	2.75
Açúcar 2kg	5.20
Sal 1kg	3.00
Óleo de Soja 1L	2.99
Óleo de Girassol 1L	7.50
Queijo 200g	6.00
Macarrão Spagetti	3.50
Lata de Atum	5.80

Faça um programa que receba do usuário quantos reais ele tem ( ex: 2.50 reais ) e analisando a lista de produtos acima, deve dizer quais produtos ele pode comprar, um por linha.

Exemplo de execução:

Com R\$ 2.00 pode comprar:

Pão 4 un.

Trigo 1kg

# Matrix

O filme matrix é conhecido por sua cena da 'chuva digital' em que várias letras verdes aleatórias descem em uma tela preta, de forma semelhante a uma tabela com colunas que escorrem com novas letras.



Faça um programa que crie uma matriz com letras aleatórias que serão preenchidas em colunas, e que a cada iteração mais letras são adicionadas e removidas (Ou alguma variação disso, o importante é lembrar um pouco a ideia do que foi feito no filme)

Basicamente uma das formas de fazer é criar uma matriz com um tamanho fixo de linhas e colunas, e então ir atravessando suas posições preenchendo as letras sobrepondo as anteriores, porém de uma forma que apenas algumas letras são preenchidas em posições consecutivas em relação à iteração anterior (linha de baixo) e a matriz é então exibida na tela a cada iteração.

## Dicas:

- É possível em javascript limpar toda a tela do terminal utilizando `console.clear()`
- Você pode utilizar a função `setInterval` para o código executar a cada X milissegundos de tempo e não travar o terminal por executar rápido demais.

## Don't Blink



De forma aproximada, um adulto pisca de 10 a 15 vezes por minuto, onde que o olho fica fechado por aproximadamente 0,2 segundos ( 200 milissegundos )

Considerando uma pessoa que queira assistir um filme de 2 horas sem perder nenhum instante, teria que assistir esse filme várias vezes, já que na primeira vez que assistir irá perder aprox. 360 segundos do filme piscando.

Considerando que existe uma variação aleatória ao piscar onde que em um minuto pode piscar de 10 a 15 vezes, para identificar o número de vezes necessário para assistir o filme sem perder nenhum instante é possível realizar uma simulação com computação.

Existem várias formas de implementar uma solução para este problema, mas aqui vai uma ideia:

Faça um programa que separe as 2 horas de um filme em uma lista de 36.000 elementos que representam os instantes que podem ter sido assistidos ou não ( $2 \text{ horas} * 60 \text{ minutos} * 60 \text{ segundos} * (1/0.2)$  ).

Utilizando um laço de repetição, atravesse essa lista de forma a registrar os instantes que foram assistidos ou não aleatoriamente baseado na estatística de que alguém pisca de 10 a 15 vezes por minuto. Após atravessar a lista preenchendo aleatoriamente se piscou ou não, informe uma estatística de quantos segundos foram perdidos piscando, quantas vezes foram piscadas, etc...

Permita que seja re-assistido o filme quantas vezes o usuário desejar, de forma a diminuir a quantidade perdida piscando conforme o filme é re-assistido.