

Exercício 1: Validação de E-mail Básico

Crie um programa que simula uma validação simples de endereço de e-mail.

1. Defina uma variável para um e-mail correto predefinido (exemplo: "usuario@dominio.com").
 2. Defina uma senha de acesso.
 3. O programa deve pedir a senha ao usuário. Use um **while** loop para garantir que o acesso só ocorra após a senha correta ser digitada.
 4. Após o login, peça ao usuário para digitar um e-mail para validação.
 5. O programa deve verificar se o e-mail digitado **contém** o caractere "@" e se o e-mail **termina** com ".com".
 6. Use uma estrutura **if** para checar as duas condições. Use o operador **and** para combinar as verificações.
 7. Se ambas as condições forem verdadeiras, exiba a mensagem: "E-mail válido!". Se uma ou ambas as condições forem falsas, exiba a mensagem: "E-mail inválido!".
 8. O programa deve perguntar se o usuário quer validar outro e-mail. Se a resposta for "sim", o processo deve se repetir. Se for "não", o programa deve encerrar.
-

Exercício 2: Validação e Formatação de Telefone

Crie um programa que recebe um número de telefone com 11 dígitos.

1. O número é considerado **inválido** se tiver 3 ou mais dígitos iguais.
2. O programa deve verificar se o número tem 11 dígitos e se todos os caracteres são números.
3. Se o número for válido, o programa deve formatá-lo para o padrão (XX)XXXXX-XXXX.
4. O programa deve imprimir o número formatado ou a mensagem de erro correspondente.

Exercício 3: Codificador e Decodificador de Frases

Crie um programa que codifica e decodifica uma frase, seguindo as regras abaixo:

Cada vogal deve ser substituída pelo número correspondente:

- $a \rightarrow 1$
- $e \rightarrow 2$
- $i \rightarrow 3$
- $o \rightarrow 4$
- $u \rightarrow 5$

O programa deve:

1. Ler uma frase digitada pelo usuário.
2. Exibir a frase **codificada**, trocando as vogais pelos números.
3. Exibir a frase **decodificada**, voltando os números às vogais originais.