

Exercício: Senha

Harvard CS50 Staff

2023-09-02

Resumo

Este exercício corresponde ao problema *password* original da disciplina Harvard CS50, em sua versão integral, traduzido e adaptado para o português pelo prof. Abrantes Araújo Silva Filho.

Sumário

1	Introdução	1
2	Senhas	2
3	Detalhes de implementação	2
3.1	Para pensar	3
4	Como testar seu código?	3
5	Como enviar seu código?	3

1 Introdução

Este exercício corresponde ao problema “*password*” da disciplina **Harvard CS50**, e deve ser feito por todos os alunos que estão estudando o conteúdo sobre *arrays*.

O objetivo deste exercício é que você pratique manipulação de strings, aprenda mais sobre a biblioteca `cctype` e trabalhe com variáveis booleanas.

A tradução e adaptação para o português foram feitas com base na versão de 2023 do exercício, conforme o [problema original](https://cs50.harvard.edu/x/2023/problems/2/password/)¹.

¹<https://cs50.harvard.edu/x/2023/problems/2/password/>

2 Senhas

Figura 1: Trecho do filme “Spaceballs”, de 1987.



Fonte: MGM Channel no YouTube²

Se o vídeo acima estiver bloqueado em seu país ou se o link não estiver mais funcionando, não se preocupe: não é obrigatório na resolução deste problema. É um trecho humorístico sobre senhas do filme “Spaceballs”.

Como todos nós sabemos, é importante utilizar senhas que não são fáceis de serem descobertas! Muitas aplicações e sites web atualmente exigem que as senhas não sejam compostas apenas de letras, mas também tenham símbolos, números e letras maiúsculas.

Neste problema você desenvolverá um programa que solicitará uma senha a um usuário e, então, fará uma validação da senha utilizando a função `validar` que você mesmo irá programar. A senha deverá conter:

- Pelo menos 1 letra maiúscula;
- Pelo menos 1 letra minúscula;
- Pelo menos 1 número; e
- Pelo menos 1 símbolo (qualquer coisa que não seja uma letra ou um número, tal como ‘!’, ‘\$’, ‘#’).

Se a senha for válida a função `validar` deverá retornar o valor booleano `TRUE`; se for inválida a função deverá retornar `FALSE`. Algumas dicas:

- A biblioteca `ctype.h` tem muitas funções que podem ser úteis em seu programa. Consulte a [documentação para aprender o que essa biblioteca pode oferecer](#)³; e
- Variáveis booleanas podem ser úteis para verificar se cada critério em um algoritmo é válido.

3 Detalhes de implementação

Utilize o *starter file* com o nome “senha.c”. Nesse arquivo a função `main` já está totalmente programada para você. Estude essa função para saber como o programa deve funcionar.

²https://www.youtube.com/watch?v=wxSTB_B0Oak

³<https://manual.cs50.io/#ctype.h>

Você deverá completar a função `validar`, que irá percorrer a senha fornecida pelo usuário (e passada como um argumento para essa função). Já que você precisará identificar se a senha conta com pelo menos uma letra maiúscula, uma letra minúscula, um número e um símbolo, talvez você queira criar uma variável booleana para cada uma dessas situações e inicializar essas variáveis com o valor `FALSE`, antes de percorrer todos os caracteres da string da senha. Se, ao percorrer a senha, você encontrar um número, por exemplo, poderá atribuir o valor `TRUE` à variável booleana para números. Se todas as variáveis booleanas forem `TRUE`, todos os critérios foram preenchidos e a senha será válida. Nesse caso a função `validar` deverá retornar `TRUE` também.

3.1 Para pensar

Quantas senhas diferentes você acredita que podem ser criadas com uma senha de 8 caracteres, usando todos os 95 caracteres ASCII imprimíveis?

4 Como testar seu código?

O comportamento esperado do seu programa, se o usuário informar senhas inválidas é o seguinte (atenção: no código abaixo a linha 3 está quebrada pois não cabe inteira nesta página, mas ela é impressa em apenas uma única linha em seu programa):

```
1 $ ./senha
2 Digite sua senha: ola
3 A senha precisa de 1 letra maiúscula, 1 letra minúscula, 1 número e
  1 símbolo!
4 $
```

```
1 $ ./senha
2 Digite sua senha: v3nt0
3 A senha precisa de 1 letra maiúscula, 1 letra minúscula, 1 número e
  1 símbolo!
4 $
```

Se o usuário digitar uma senha válida, o comportamento esperado é o seguinte:

```
1 $ ./senha
2 Digite sua senha: v3nt0!
3 Sua senha é válida!
4 $
```

Lembre-se também de que seu código deve seguir todas as normas de estilo de programação C da disciplina Harvard CS50: [Harvard CS50 C Style Guide](https://cs50.readthedocs.io/style/c/)⁴.

5 Como enviar seu código?

Utilize o starter file “`senha.c`” (mantenha esse padrão de nome), preencha as informações de identificação e envie o arquivo no Autolab, no exercício denominado “Senha”.

⁴<https://cs50.readthedocs.io/style/c/>