

Exercício: S3m v0g61s

Harvard CS50 Staff

2023-09-02

Resumo

Este exercício corresponde ao problema *no vowels* 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	l33tsp36k	1
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 “*no vowels*” 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 a passar argumentos de linha de comando para seu programa, e que escreva o programa do zero.

A tradução e adaptação para o português foram feitas com base na versão de 2023 do exercício, conforme o [problema original](#)¹.

2 l33tsp36k

Se você já navegou na internet, já deve ter visto o “[leetspeak](#)”² (ou “l33tsp36k” para nossos propósitos!), que envolve a substituição de letras por símbolos, onde esses símbolos se assemelham de alguma forma a seus equivalentes alfabéticos. Por exemplo: a palavra “escola” pode ser escrita como “3sc0l6”. Neste exercício você escreverá um programa para substituir certas vogais por dígitos!

Neste exercício você começará com um código praticamente vazio, somente com algumas bibliotecas comuns. A função `main` também estará vazia. Seu trabalho será criar um programa que **receba 1 argumento via linha de comando**, a palavra a ser convertida, e imprima a palavra convertida no terminal (substituindo as vogais por números pré-determinados).

Algumas dicas:

¹<https://cs50.harvard.edu/x/2023/problems/2/no-vowels/>

²<https://en.wikipedia.org/wiki/Leet>

- Note que a função `main` usa dois parâmetros: um número inteiro, `argc`, e uma string, `argv`. Esses dois parâmetros são essenciais no processamento de argumentos passados via linha de comando, você terá que usá-los em seu programa; e
- Verifique se o usuário informou o número correto de argumentos da linha de comando antes de fazer qualquer coisa na função `main`.



Fonte: Harvard CS50

3 Detalhes de implementação

Utilize o *starter file* com o nome “`sem_vogais.c`”. Nesse arquivo já incluímos as bibliotecas comuns que você utilizará, e já declaramos a função `main` com os parâmetros necessários para você, mas todo o resto está em branco. Você deve fazer criar o programa de acordo com as seguintes especificações:

- O programa deve aceitar um, e apenas um, argumento de linha de comando, que será a palavra a ser convertida;
- Se o usuário não passar nenhum argumento de linha comando ou se passar mais de um argumento, seu programa deverá imprimir uma mensagem de erro de sua escolha (através de `printf`) e fazer com que a função `main` retorne um valor de 1 (que geralmente indica algum erro de execução) imediatamente, ou seja, seu programa vai retornar 1 e terminar;
- Seu programa deve ter, obrigatoriamente, uma função chamada de `trocar`, que recebe uma string como argumento e retorna uma outra string com as vogais substituídas por números. Além dessa função obrigatória seu programa pode conter outras funções se você achar necessário;
- A função `trocar` deve fazer as seguintes substituições: “a” por 6; “e” por 3; “i” por 1; “o” por 0; e “u” por v. Somente testaremos seu código para vogais minúsculas, sem nenhum tipo de acentos;
- A string que será o argumento da função `trocar` será a entrada que o usuário digitou na linha de comando, armazenada em `argv[1]`;
- Seu programa deverá imprimir a palavra convertida, seguida de uma quebra de linha;
- Estude como usar a [sentença `switch case`](https://cs50.readthedocs.io/style/c/#switches)³, pois isso poderá facilitar bastante a criação da função `trocar`.

³<https://cs50.readthedocs.io/style/c/#switches>

3.1 Para pensar

Por que você iria querer usar argumentos de linha de comando para um programa ao invés de solicitar as informações do usuário com o uso de funções como `get_int` ou `get_string`?

4 Como testar seu código?

O comportamento esperado do seu programa, se o usuário não informar nenhuma palavra como argumento na linha de comando, deve ser o seguinte:

```
1 $ ./sem_vogais
2 Uso: ./sem_vogais palavra
3 $
```

Atenção: apesar de parecer que o programa só imprimiu a frase com as instruções de uso (linha 2 no programa acima), ele fez mais do que isso: ele retornou o código de erro “1” para o terminal de comandos. Para poder ver qual foi o código que um programa retornou para o terminal, utilize comando “`echo $?`” imediatamente após o fim do programa:

```
1 $ ./sem_vogais
2 Uso: ./sem_vogais palavra
3 $ echo $?
4 1
5 $
```

Se o usuário informar mais de um único argumento na linha de comando, seu programa também deve imprimir uma mensagem de erro e retornar o código 1:

```
1 $ ./sem_vogais casa aviao
2 Uso: ./sem_vogais palavra
3 $
```

Se o usuário informar apenas um único argumento na linha de comando, seu programa deve retornar a string alterada:

```
1 $ ./sem_vogais pseudocodigo
2 ps3vd0c0dlg0
3 $
```

Lembre-se que, para facilitar, seu programa só precisa converter as vogais minúsculas (a, e, i, o, u), sem qualquer tipo de acentuação.

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 “`sem_vogais.c`” (mantenha esse padrão de nome), preencha as informações de identificação e envie o arquivo no Autolab, no exercício denominado “S3m v0g61s”.

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