

# PSET: Legibilidade (*Readability*)

Harvard CS50 Staff

2023-09-20

## Resumo

Esta atividade corresponde ao PSET *Readability* (legibilidade) original da disciplina Harvard CS50, em sua versão integral, traduzido e adaptado para o português pelo prof. Abrantes Araújo Silva Filho.

## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Legibilidade (<i>Readability</i>)</b>	<b>2</b>
<b>3</b>	<b>Detalhes de implementação</b>	<b>4</b>
3.1	Obtendo a entrada do usuário . . . . .	4
3.2	Letras . . . . .	4
3.3	Palavras . . . . .	5
3.4	Frases . . . . .	5
3.5	Juntando tudo . . . . .	6
<b>4</b>	<b>Passo a passo</b>	<b>6</b>
<b>5</b>	<b>Como testar seu código?</b>	<b>7</b>
<b>6</b>	<b>Como enviar seu código?</b>	<b>8</b>

## 1 Introdução

Este exercício corresponde ao PSET “*Readability*” (nível de leitura, legibilidade) 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 e arrays, fazendo acesso direto à diversas posições do array através de seu índice. Também aprenderá mais sobre as bibliotecas `cctype.h`, `math.h` e `string.h`.

A tradução e adaptação para o português foram feitas com base na versão de 2023 desta atividades, conforme o [PSET original](https://cs50.harvard.edu/x/2023/psets/2/readability/)<sup>1</sup>.

---

<sup>1</sup><https://cs50.harvard.edu/x/2023/psets/2/readability/>

## 2 Legibilidade (*Readability*)

Para este problema, você implementará um programa que calcula o nível de legibilidade aproximado que é necessário para que uma pessoa possa ler e compreender um determinado texto, em inglês, como ilustrado no exemplo abaixo. Note que: a) como o programa a ser desenvolvido está ajustado para o cálculo da legibilidade de textos em inglês, os exemplos e testes deverão utilizar essa língua<sup>2</sup>; e b) o texto na linha 2 do exemplo abaixo é digitado em uma única linha, apesar de parecer que está em duas linhas por motivos de espaço):

```
1 ./legibilidade
2 Texto: Congratulations! Today is your day. You're off to Great
  Places! You're off and away!
3 Grau 3
4 $
```

De acordo com o site [Scholastic](#)<sup>3</sup>, “*Charlotte’s Web*”, de E. B. White, está entre o nível de leitura do segundo e quarto ano, e “*The Giver*”, de Lois Lowry, está entre o nível de leitura do oitavo ao décimo segundo ano. Mas o que significa, de fato, um livro estar em um determinado nível de leitura?

Bem, em muitos casos, um especialista humano pode ler um livro e decidir o ano escolar (ou seja, ano letivo) para o qual acha que o livro é mais apropriado. Mas um algoritmo provavelmente também poderia determinar isso!

Então, quais tipos de características são típicas de níveis de leitura mais altos? Bem, palavras mais longas provavelmente se correlacionam com níveis de leitura mais altos. Da mesma forma, frases mais longas provavelmente também têm relação com níveis de leitura mais altos.

Vários “testes de legibilidade” foram desenvolvidos ao longo dos anos para definir fórmulas para calcular o nível de leitura de um texto. Um desses testes de legibilidade é o **Índice Coleman-Liau**. O índice Coleman-Liau, de um texto em inglês, projetado para indicar o ano escolar (nos EUA) necessário para compreender um determinado texto tem a fórmula

$$\text{índice} = 0.0588 \times L - 0.296 \times S - 15.8 \quad (1)$$

onde  $L$  é o número médio de letras por 100 palavras no texto, e  $S$  é o número médio de sentenças (frases) por 100 palavras no texto.

Vamos escrever um programa chamado “legibilidade”, que pega um texto (em inglês) e determina seu nível de leitura. Por exemplo, se o usuário digitar uma linha de texto do “*Dr. Seuss*”, o programa deve se comportar da seguinte maneira:

```
1 ./legibilidade
2 Texto: Congratulations! Today is your day. You're off to Great
  Places! You're off and away!
3 Grau 3
4 $
```

O texto que o usuário inseriu tem 65 letras, 4 sentenças (frases) e 14 palavras. 65 letras por 14 palavras tem uma média de cerca de 464,29 letras a cada 100 palavras  $((65/14) \times 100 = 464,29)$ . E 4 frases por 14 palavras têm uma média de cerca de 28,57 frases a cada 100 palavras  $((4/14) \times 100 = 28,57)$ . Inserindo na fórmula Coleman-Liau e arredondando para o número inteiro mais próximo, obtemos uma resposta de 3  $((0,0588 \times 464,29) - (0,296 \times 28,57) - 15,8 \approx 3)$ . Então, este trecho está em um nível de leitura de terceiro ano.

Vamos tentar outro:

<sup>2</sup>Se você tiver curiosidade e quiser saber mais sobre como calcular a legibilidade para textos em português, além de aprender sobre outros índices de legibilidade, visite <https://legibilidade.com>.

<sup>3</sup><https://www.scholastic.com/teachers/teaching-tools/collections/guided-reading-book-lists-for-every-level.html>

```
1 ./legibilidade
2 Texto: Harry Potter was a highly unusual boy in many ways. For one
  thing, he hated the summer holidays more than any other time of
  year. For another, he really wanted to do his homework, but was
  forced to do it in secret, in the dead of the night. And he also
  happened to be a wizard.
3 Grau 5
4 $
```

Este texto tem 214 letras, 4 frases e 56 palavras. Isso resulta em cerca de 382,14 letras a cada 100 palavras e 7,14 frases a cada 100 palavras. Inserindo na fórmula Coleman-Liau, obtemos um nível de leitura do quinto ano.

Conforme o número médio de letras e palavras por frase aumenta, o índice Coleman-Liau atribui ao texto um nível de leitura mais alto. Se você pegar este parágrafo, por exemplo, que tem palavras e frases mais longas do que os dois exemplos anteriores, a fórmula atribuiria ao texto um nível de leitura do décimo segundo ano.

```
1 ./legibilidade
2 Texto: As the average number of letters and words per sentence
  increases, the Coleman-Liau index gives the text a higher readin
  level. If you were to take this paragraph, for instance, which has
  longer words and sentences than either of the prior two examples,
  the formula would give the text an twelfth-grade reading level.
3 Grau 12
4 $
```

As figuras 1 e 2, abaixo, ilustram uma execução real do programa final.

Figura 1: Avaliação da legibilidade

```
$ ./readability
Text: One fish. Two fish. Red fish. Blue fish.
Before Grade 1
$ ./readability
Text: Congratulations! Today is your day. You're off to Great Places! You're off and away!
Grade 3
$ ./readability
Text: Harry Potter was a highly unusual boy in many ways. For one thing, he hated the summer holiday
s more than any other time of year. For another, he really wanted to do his homework, but was forced
to do it in secret, in the dead of the night. And he also happened to be a wizard.
Grade 5
$
```

Fonte: Harvard CS50 Staff

Figura 2: Avaliação da legibilidade

```
$ ./readability
Text: When he was nearly thirteen, my brother Jem got his arm badly broken at the elbow. When it hea
led, and Jem's fears of never being able to play football were assuaged, he was seldom self-consciou
s about his injury. His left arm was somewhat shorter than his right; when he stood or walked, the b
ack of his hand was at right angles to his body, his thumb parallel to his thigh.
Grade 8
$ ./readability
Text: A large class of computational problems involve the determination of properties of graphs, dig
raphs, integers, arrays of integers, finite families of finite sets, boolean formulas and elements o
f other countable domains.
Grade 16+
$
```

Fonte: Harvard CS50 Staff

### 3 Detalhes de implementação

Projete e implemente o programa, “legibilidade”, que calcula o índice Coleman-Liau de um texto em inglês.

- Implemente seu programa em um arquivo chamado “legibilidade.c”;
- Seu programa deve solicitar uma `string` de texto ao usuário, usando `get_string`;
- Seu programa deve contar o número de letras, palavras e frases no texto. Você pode presumir que uma letra é qualquer caractere minúsculo de “a” a “z” ou qualquer caractere maiúsculo de “A” a “Z”. Qualquer sequência de caracteres separados por espaços deve contar como uma palavra. Considere que qualquer ocorrência de um ponto, ponto de exclamação ou ponto de interrogação indica o final de uma frase;
- Seu programa deve imprimir como saída, exatamente “Grau X”, onde X é o nível de série calculado pela fórmula Coleman-Liau, arredondado para o número inteiro mais próximo. Se o número do índice resultante for 16 ou superior (equivalente ou superior a um nível de leitura de graduação no último ano), seu programa deve exibir “Grau 16+” em vez de fornecer o número exato do índice. Se o número do índice for menor que 1, seu programa deve exibir “Abaixo do Grau 1”.

#### 3.1 Obtendo a entrada do usuário

Em primeiro lugar vamos escrever algum código em C que obtenha a entrada de texto do usuário e a imprima de volta. Especificamente, implemente na função `main` do programa, alguma coisa que mostre o prompt “Texto: ”, usando a função `get_string` e, depois, imprima o mesmo texto que o usuário informou usando a `printf`. Lembre-se que, se você usar funções adicionais de outras bibliotecas, precisará incluir as *header files* correspondentes utilizando o `#include`.

O programa deve se comportar conforme o descrito abaixo (o texto que o usuário digitou na linha 2 deve ser impresso, sem modificações na linha 3):

```
1 ./legibilidade
2 Texto: In my younger and more vulnerable years my father gave me
   some advice that I've been turning over in my mind ever since.
3 In my younger and more vulnerable years my father gave me some
   advice that I've been turning over in my mind ever since.
4 $
```

#### 3.2 Letras

Agora que você já conseguiu obter o input do usuário (o texto que ele digitar), vamos começar a análise fazendo a contagem das letras. Considere como letras apenas os caracteres a–z ou A–Z, mas não considere como letras sinais de pontuação, números ou outros símbolos.

Você deve criar uma função chamada `contar_letras` que recebe um único argumento, a `string` de texto informada pelo usuário, e que retorna um `int`, o número de letras no texto. Não se esqueça de colocar o protótipo da função acima da função `main`, que deve se parecer com o seguinte:

```
int contar_letras(string texto);
```

Agora faça uma alteração em seu programa para que, ao invés de imprimir o texto informado pelo usuário, agora ele imprima o número de letras no texto. Por exemplo:

```
1 ./legibilidade
2 Texto: Alice was beginning to get very tired of sitting by her
  sister on the bank, and of having nothing to do: once or twice she
  had peeped into the book her sister was reading, but it had no
  pictures or conversations in it, "and what is the use of a book,"
  thought Alice "without pictures or conversation?"
3 235 letras
4 $
```

Dica: na biblioteca `ctype.h` existe uma função que você pode achar útil para fazer a identificação das letras. Consulte a documentação dessa biblioteca em <https://manual.cs50.io>. Se você utilizar essa biblioteca não se esqueça de incluí-la no começo de seu programa.

### 3.3 Palavras

O índice Coleman-Liau se preocupa não apenas com o número de letras mas, também, com o número de palavras em uma frase. Para o propósito deste problema, vamos considerar qualquer sequência de caracteres separados por um espaço como uma palavra (note que uma palavra hifenizada, como “*sister-in-law*”, deve ser considerada uma única palavra).

Adicione ao `legibilidade.c`, uma função chamada `contar_palavras` que recebe como argumento uma `string` de texto, e que retorna um `int`, o número de palavras nesse texto. Certifique-se de adicionar o protótipo da função no começo do seu programa (para que a função `main` reconheça essa função).

Teste o seu programa fazendo com que ele imprima, também, a quantidade de palavras no texto. Você pode assumir que uma frase:

- Contará com, pelo menos, uma palavra;
- Não vai começar ou terminar com um espaço; e
- Não terá vários espaços lado a lado.

É claro que você pode tentar uma solução que tolerará vários espaços entre palavras ou, de fato, sem palavras! O programa deve executar conforme o exemplo a seguir:

```
1 ./legibilidade
2 Texto: It was a bright cold day in April, and the clocks were
  striking thirteen. Winston Smith, his chin nuzzled into his breast
  in an effort to escape the vile wind, slipped quickly through the
  glass doors of Victory Mansions, though not quickly enough to
  prevent a swirl of gritty dust from entering along with him.
3 250 letras
4 55 palavras
5 $
```

### 3.4 Frases

A última informação sobre a qual a fórmula Coleman-Liau se preocupa, além do número de letras e palavras, é o número de frases. Determinar o número de frases pode ser surpreendentemente complicado. Você pode primeiro imaginar que uma frase é apenas qualquer sequência de caracteres que termina com um ponto, mas é claro que as frases podem terminar com um ponto de exclamação ou um ponto de interrogação também. Mas, é claro, nem todos os períodos significam necessariamente que a sentença acabou. Por exemplo, considere a frase abaixo.

*Mr. and Mrs. Dursley, of number four Privet Drive, were proud to say that they were perfectly normal, thank you very much.*

É apenas uma frase, mas há três pontos! Nesta atividade você deve ignorar essa sutileza: considere qualquer sequência de caracteres que termina com um “.” ou um “?” ou uma “!” como uma frase (a frase do exemplo anterior deve ser contada como três frases). Na prática a detecção de limites de frases precisa ser um pouco mais inteligente para lidar com esses casos, mas não nos preocuparemos com isso por enquanto.

Adicione ao `legibilidade.c` uma função chamada `contar_palavras`, que recebe como argumento uma `string` de texto, e que retorna um `int`, o número de frases nesse texto. Certifique-se de adicionar o protótipo da função no começo de seu programa (para que a função `main` reconheça essa função).

Teste seu programa e verifique se ele está funcionando conforme o seguinte exemplo:

```
1 ./legibilidade
2 Texto: When he was nearly thirteen, my brother Jem got his arm
   badly broken at the elbow. When it healed, and Jem's fears of never
   being able to play football were assuaged, he was seldom
   self-conscious about his injury. His left arm was somewhat shorter
   than his right; when he stood or walked, the back of his hand was
   at right angles to his body, his thumb parallel to his thigh.
3 295 letras
4 70 palavras
5 3 frases
6 $
```

### 3.5 Juntando tudo

Agora é hora de juntar todas as peças! Lembre-se de que o índice Coleman-Liau é calculado usando a fórmula especificada na Equação 1. Modifique seu programa para que ele imprima o grau de legibilidade (“Grau 2”, “Grau 8”, “Grau 16+”, “Abaixo do Grau 1”, etc.). Lembre-se de arredondar o resultado para o número inteiro mais próximo (talvez você ache útil a função `round`, da biblioteca `math.h`).

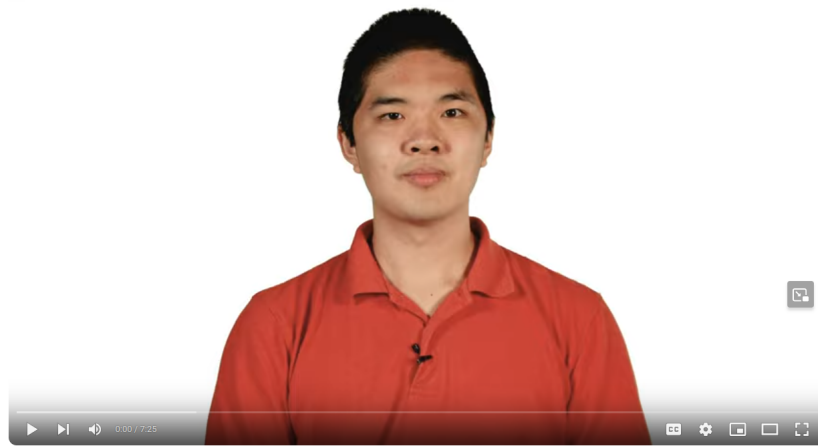
Algumas dicas:

- Lembre-se de que a documentação da função `round`, da biblioteca `math.h`, pode ser encontrada em <https://manual.cs50.io>; e
- Lembre-se de que, ao dividir valores `int` em C, o resultado também será um `int`, descartando todos os algarismos depois do ponto decimal. Dito de outra forma, o resultado será “truncado”. Talvez seja necessário que você realize a alteração do tipo de dados (*cast*) antes de realizar divisões ao calcular os valores de  $L$  e  $S$ .

Não se esqueça também de que se o número de índice resultante for 16 ou mais (equivalente a ou maior do que um nível de leitura de graduação sênior), seu programa deve imprimir “Grau 16+” ao invés de mostrar o índice exato. Se o número do índice for menor que 1, seu programa deve produzir “Abaixo do Grau 1”.

## 4 Passo a passo

Se você precisar de ajuda para começar, assista este vídeo:



Fonte: Harvard CS50 Staff<sup>4</sup>

## 5 Como testar seu código?

Utilize as **frases de teste** (em inglês) que estão listadas no arquivo “`frases.txt`”, que acompanha este documento PDF. Você deve copiar e colar cada frase, exatamente como está escrita, e verificar se o seu programa retorna a resposta correta. As frases que estão no arquivo de teste são as seguintes:

- *One fish. Two fish. Red fish. Blue fish.* (Abaixo do Grau 1)
- *Would you like them here or there? I would not like them here or there. I would not like them anywhere.* (Grau 2)
- *Congratulations! Today is your day. You're off to Great Places! You're off and away!* (Grau 3)
- *Harry Potter was a highly unusual boy in many ways. For one thing, he hated the summer holidays more than any other time of year. For another, he really wanted to do his homework, but was forced to do it in secret, in the dead of the night. And he also happened to be a wizard.* (Grau 5)
- *In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.* (Grau 7)
- *Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, “and what is the use of a book,” thought Alice “without pictures or conversation?”* (Grau 8)
- *When he was nearly thirteen, my brother Jem got his arm badly broken at the elbow. When it healed, and Jem's fears of never being able to play football were assuaged, he was seldom self-conscious about his injury. His left arm was somewhat shorter than his right; when he stood or walked, the back of his hand was at right angles to his body, his thumb parallel to his thigh.* (Grau 8)
- *There are more things in Heaven and Earth, Horatio, than are dreamt of in your philosophy.* (Grau 9)
- *It was a bright cold day in April, and the clocks were striking thirteen. Winston Smith, his chin nuzzled into his breast in an effort to escape the vile wind, slipped quickly through the glass doors of Victory Mansions, though not quickly enough to prevent a swirl of gritty dust from entering along with him.* (Grau 10)
- *A large class of computational problems involve the determination of properties of graphs, digraphs, integers, arrays of integers, finite families of finite sets, boolean formulas and elements of other countable domains.* (Grau 16+)

---

<sup>4</sup><https://www.youtube.com/watch?v=AOVyZEh9zgE>

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/)<sup>5</sup>.

## 6 Como enviar seu código?

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

---

<sup>5</sup><https://cs50.readthedocs.io/style/c/>