

Readability

Introdução

De acordo com o site [Scholastic](#), E."Charlotte's Web" de B. White é indicado para alunos de 7 a 10 anos (2nd to 4th grade [1]) e "The Giver" de Lois Lowry é indicado para alunos de 13 a 18 anos (8th to 12th grade). O que significa ter um quinto nível "grade" de leitura?

Bem, em muitos casos, um humano pode ler um livro e classificar em que *grade* ele acha que o livro deveria estar. Mas você poderia pensar em um algoritmo que tenta realizar a mesma tarefa de classificar o nível de um texto.

Então, que tipo de características um leitor de ensino médio deveria ter? Bem, palavras longas provavelmente devem estar relacionadas com leitores mais experientes. Do mesmo modo que sentenças longas devem ser mais fáceis para estes leitores. Vários **testes de readability** foram desenvolvidos por anos para dar um modelo computável do nível de leitura de um texto.

Um destes testes é o índice Coleman-Liau para textos em inglês. Este teste indica em que *grade* um aluno deveria estar. Sua fórmula é

$$S\&I = 0.0588 * L - 0.296 * S - 15.8$$

onde L é a o número médio de letras a cada 100 palavras no texto e S é a média do número de sentenças a cada 100 palavras no texto.

Vamos escrever um programa chamado **readability** que recebe um texto como entrada e determina o nível de leitura. Por exemplo,

```
$ ./readability
Text: Congratulations! Today is your day. You're off to Great Places! You're off
and away!
Grade 3
```

O texto possui 65 letras, 4 sentenças e 14 palavras. Então, 65 letras em 14 palavras dá uma média de 464.29 letras para cada 100 palavras. Do mesmo modo, 4 sentenças e 14 palavras dá uma média de 28.57 sentenças por 100 palavras. Substituindo os valores na fórmula de Coleman-Liau e arredondando para o inteiro mais próximo, nós obtemos a resposta **3**.

Outro exemplo

```
$ ./readability
Text: 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.
Grade 5
```

O texto possui 214 letras, 4 sentenças e 56 palavras. Isto nos dá 328.14 letras e 7.14 sentenças para cada 100 palavras. Substituindo na fórmula, obtemos um nível 5.

Conforme aumenta o número de letras em uma palavra e palavras por sentença, o índice de Coleman-Liau aumenta. Se você ler um parágrafo com palavras mais longas e mais palavras por frase, a fórmula lhe dará um valor mais alto.

Por exemplo

```
$ ./readability
Text: As the average number of letters and words per sentence increases, the
Coleman-Liau index gives the text a higher reading 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 eleventh grade reading
level.
Grade 11
```

Especificação do Problema

Projete e implemente o programa `readability` que calcula o índice de Coleman-Liau de um texto.

- Implemente seu programa em um arquivo chamado `readability.c` no diretório `~/pset2/readability`.
- Seu programa precisa solicitar (*prompt*) uma *string* de texto ao usuário (usando `get_string`).
- Seu programa deveria contar o número de letras, palavras e frases no texto. Você deve assumir que **letra** é qualquer caractere minúsculo de `a` a `z` ou maiúsculo de `A` a `Z`. Uma **palavra** é qualquer sequência de caracteres separadas por espaço. Uma frase é qualquer sequência de caracteres separadas por ponto, exclamação ou interrogação. Note que não estamos preocupados em considerar (Dr.) como uma frase.
- Seu programa deveria exigir uma saída `"Grade X"` onde `X` é o nível *grade* calculado pela fórmula Coleman-Liau arredondada para o inteiro mais próximo.
- Se o resultado é maior ou igual a 16, (o que equivale ao nível de um aluno de graduação), seu programa deve exibir `"Grade 16+"` em vez de fornecer o número exato. Se o número é menor que 1, seu programa deveria exibir `"Before Grade 1"`.

Solicitando dados ao usuário

Vamos escrever um código em C que simplesmente pega um texto fornecido pelo usuário e exibe o mesmo texto na tela. Especificamente, escreva o código em um arquivo chamado `readability.c` de tal modo que o usuário execute o programa e seja solicitado (através de um aviso `"Text:"`) a digitar um texto.

Seu programa deveria se comportar como abaixo:

```
$ ./readability
Text: In my younger and more vulnerable years my father gave me some advice that
I've been turning over in my mind ever since.
```

```
In my younger and more vulnerable years my father gave me some advice that I've
been turning over in my mind ever since.
```

Letras

Agora que você já está obtendo o texto do usuário, vamos analisar a entrada de dados e contar o número de letras. Modifique o `readability.c` para que em vez de exibir a frase digitada, mostrar a quantidade de letras do texto.

Seu programa deveria se comportar como abaixo:

```
$ ./readability
Text: 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?"
235 letter(s)
```

Letras podem ser caracteres alfabéticos maiúsculos ou minúsculos.

Você pode consultar pela biblioteca **cctype.h** por funções que podem ajudar. Tente, escrever uma função separada como `contar_letras` (ou `count_letters`) para manter seu código organizado.

Palavras

O índice de Coleman-Liau precisa levar em conta não só o número de letras, mas também de palavras em uma frase. Para este propósito, você deverá qualquer sequência de caracteres separados por espaço como uma palavra (então, palavras separadas por hífen como "sistem-in-law" serão consideradas como uma única palavra em vez de três).

Modifique o `readability.c` de modo que ele exiba, também, a quantidade de palavras do texto digitado.

Você pode assumir que uma frase não vai iniciar nem terminar com um espaço em branco e que não haverá múltiplos espaços em branco entre palavras.

Seu programa deveria se comportar como abaixo:

```
$ ./readability
Text: 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.
250 letter(s)
55 word(s)
```

Frases

A última peça para calcular o índice de Coleman-Liau é o número de palavras por sentença. Para determinar o número de sentenças em um texto, você deve considerar uma sequência de caracteres separadas por sinais de pontuação (ponto, exclamação e interrogação). Note que nem todo ponto indica, de fato, uma frase. Por exemplo

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

Possui uma única sentença e não três. Para o problema **readability**, você pode ignorar esta situação. Na prática, detectar sentenças é um pouco mais complicado do que o que você precisa fazer no problema em questão.

Modifique o **redability.c** para exibir, também, o número de sentenças no texto.

Seu programa deveria se comportar como abaixo:

```
$ ./readability
Text: 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.
295 letter(s)
70 word(s)
3 sentence(s)
```

Juntando tudo

Agora, é hora de juntar tudo. Lembre-se que o índice de Coleman-Liau é dado por: $SI = 0.0588 * L - 0.296 * S - 15.8$

onde L é a o número médio de letras a cada 100 palavras no texto e S é a média do número de sentenças a cada 100 palavras no texto.

Modifique o **readability.c** para em vez de exibir o número de letras, palavras e frases, mostrar o `_grade_` de acordo com o índice de Coleman-Liau (por exemplo **Grade 2** ou **Grade 8**). Você precisa garantir que o resultado foi arredondado para o inteiro mais próximo.

Se o resultado for maior ou igual a 16, seu programa deveria exibir "**Grade 16+**" e se for menor que 1 deveria exibir "**Before Grade 1**".

Dica

- A biblioteca de matemática **math.h** possui uma função para arredondar números.
- Não esqueça que operações aritméticas envolvendo números inteiros, **int**, sempre resultam em um número inteiro e que a parte decimal será descartada.

Testando seu código

Tente executar seu código com os textos abaixo

- One fish. Two fish. Red fish. Blue fish. (Before Grade 1)
- Would you like them here or there? I would not like them here or there. I would not like them anywhere. (Grade 2)
- Congratulations! Today is your day. You're off to Great Places! You're off and away! (Grade 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. (Grade 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. (Grade 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?" (Grade 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. (Grade 8)
- There are more things in Heaven and Earth, Horatio, than are dreamt of in your philosophy. (Grade 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. (Grade 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. (Grade 16+)

Execute o comando abaixo para verificar a **corretude** do seu código. Mas tente compilar e testar antes de executar o comando

```
check50 cs50/problems/2020/x/readability
```

Execute o comando abaixo para garantir a **estilização** do código

```
style50 readability.c
```

Enviando seu código

Execute o comando abaixo, logando com seu **nome de usuário** do GitHub, para enviar seu código. Por questões de segurança, asteriscos serão exibidos em vez dos caracteres da sua senha

```
submit50 cs50/problems/2020/x/readability
```

Referências

[1] [Educação nos Estados Unidos](#)