Este é o CS50

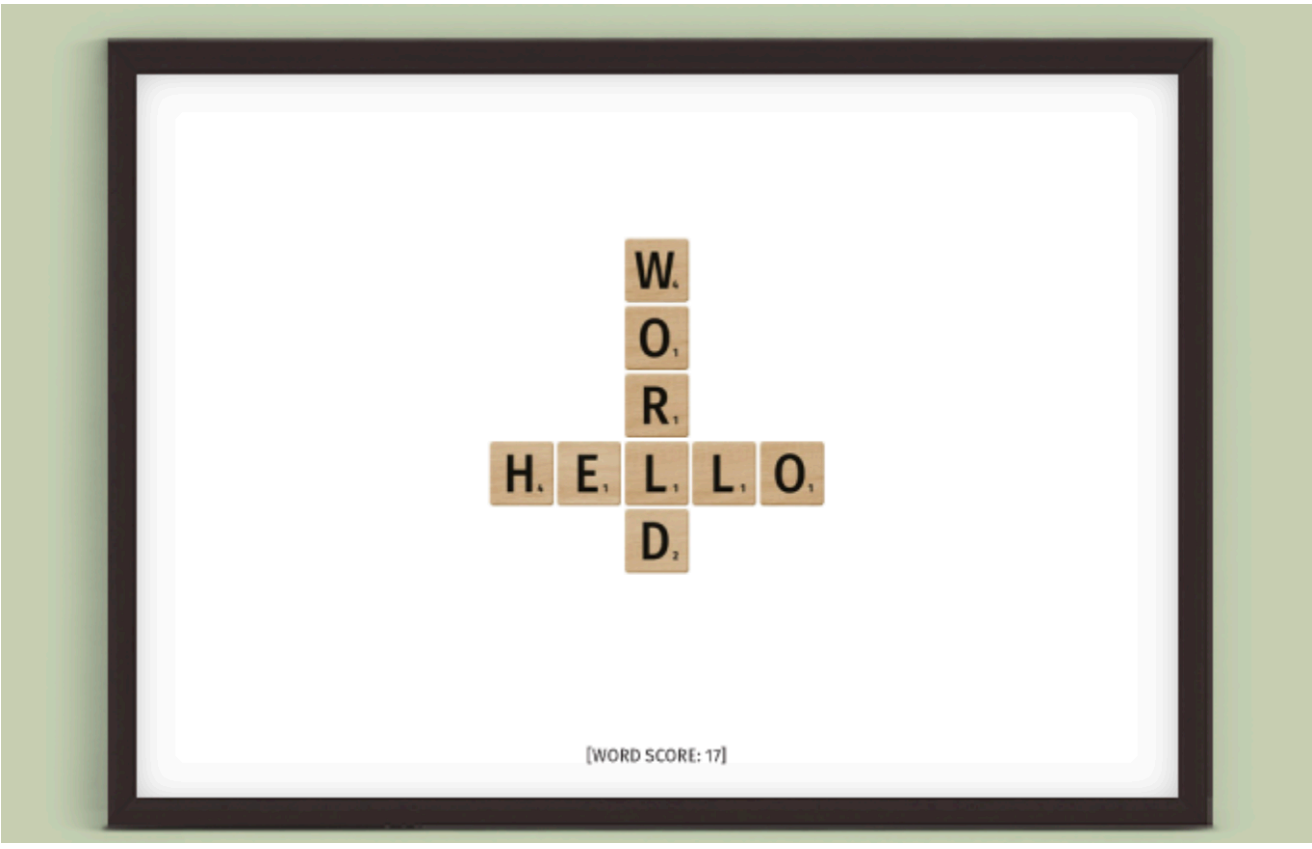
Introdução à Ciência da Computação (CS50)

OpenCourseWare

Doar (https://cs50.harvard.edu/donate)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.facebook.com/dmalan) (https://github.com/dmalan)
(https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/)
(https://www.reddit.com/user/davidjmalan)
(https://www.threads.net/@davidjmalan) (https://twitter.com/davidjmalan)

Scrabble



Problema a resolver

No jogo Scrabble (https://en.wikipedia.org/wiki/Scrabble) , os jogadores formam palavras para marcar pontos, e a pontuação total é a soma dos valores de cada letra na palavra.

UM	B	C	D	E	F	G	H	EU	J	K	L	M	N	O	P	Q	R
1	3	3	2	1	4	2	4	1	8	5	1	3	1	1	3	10	1

Por exemplo, se quiséssemos pontuar a palavra "CODE", notaríamos que o 'C' vale 3 pontos, o 'O' vale 1 ponto, o 'D' vale 2 pontos e o 'E' vale 1 ponto. Somando esses valores, obtemos que "CODE" vale 7 pontos.

Em um arquivo chamado `<nome_do_arquivo>` `scrabble.c` em uma pasta chamada ` `scrabble` <nome_da_pasta>`, implemente um programa em C que determine o vencedor de um jogo curto semelhante ao Scrabble. Seu programa deve solicitar a entrada do jogador duas vezes: uma vez para o "Jogador 1" inserir sua palavra e outra vez para o "Jogador 2" inserir a sua. Em seguida, dependendo de qual jogador obtiver a maior pontuação, seu programa deve imprimir "Jogador 1 venceu!", "Jogador 2 venceu!" ou "Empate!" (caso os dois jogadores obtenham a mesma pontuação).

Demonstração

```
Player 1: Question?
Player 2: Question!
Tie!
$ ./scrabble
Player 1: red
Player 2: wheelbarrow
Player 2 wins!
$ ./scrabble
Player 1: COMPUTER
Player 2: science
Player 1 wins!
$
```

Recorded with **asciinema**

Conselho

Clique nos botões abaixo para ler algumas dicas!

▼ **Escreva um código que você sabe que será compilado.**

```c
#include <ctype.h>
#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{

}
```

Note que você agora incluiu alguns arquivos de cabeçalho que lhe darão acesso a funções que podem ajudá-lo a resolver esse problema.

▼ **Escreva um pseudocódigo antes de escrever mais código.**

Se não tiver certeza de como resolver o problema em si, divida-o em problemas menores que você provavelmente conseguirá resolver primeiro. Por exemplo, este problema, na verdade, se resume a apenas alguns problemas:

1. Solicitar ao usuário duas palavras.
2. Calcule a pontuação de cada palavra.
3. Imprima o vencedor

Vamos escrever um pseudocódigo como comentários para lembrá-lo de fazer exatamente isso:

```c
#include <ctype.h>
#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // Prompt the user for two words

    // Compute the score of each word

    // Print the winner
}
```

Alguns problemas em listas de exercícios, como esta, podem conter spoilers (como o próximo) que revelam a solução completa. Embora seja permitido usar esse código, recomendamos fortemente que você tente resolver tudo sozinho primeiro! Os outros problemas da lista não terão esse tipo de passo a passo e, normalmente, o problema que contém o "spoiler completo" é uma versão preparatória do problema maior que você precisará resolver posteriormente.

▼ **Converta o pseudocódigo em código.**

Primeiro, considere como você poderia solicitar ao usuário duas palavras. Lembre-se de que `get_string` a função `success`, da biblioteca CS50, pode solicitar ao usuário uma string.

```
#include <ctype.h>
#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // Prompt the user for two words
    string word1 = get_string("Player 1: ");
    string word2 = get_string("Player 2: ");

    // Compute the score of each word

    // Print the winner
}
```

Em seguida, considere como calcular a pontuação de cada palavra. Como o mesmo algoritmo de pontuação se aplica a ambas as palavras, você tem uma boa oportunidade para *abstração* . Aqui, definiremos uma função chamada `score` `compute_score` que recebe uma string, chamada `word` `word`, como entrada e retorna `word` a pontuação de `word` como um `std::string` `int`.

```
#include <ctype.h>
#include <cs50.h>
#include <stdio.h>
#include <string.h>

int compute_score(string word);

int main(void)
{
    // Prompt the user for two words
    string word1 = get_string("Player 1: ");
    string word2 = get_string("Player 2: ");

    // Compute the score of each word
    int score1 = compute_score(word1);
    int score2 = compute_score(word2);

    // Print the winner
}

int compute_score(string word)
{
    // Compute and return score for word
}
```

Agora, vamos à implementação `compute_score`. Para calcular a pontuação de uma palavra, você precisa saber o valor em pontos de cada letra na palavra. Você pode associar letras e seus valores em pontos a um *array* . Imagine um array de 26 `int` letras, chamado `POINTS` `A`, no qual o primeiro número é o valor em pontos de 'A', o segundo número é o valor em pontos de 'B', e

assim por diante. Ao declarar e inicializar esse array fora de qualquer função específica, você garante que ele seja acessível a qualquer função, incluindo `score` `compute_score`.

```
#include <ctype.h>
#include <cs50.h>
#include <stdio.h>
#include <string.h>

// Points assigned to each letter of the alphabet
int POINTS[] = {1, 3, 3, 2, 1, 4, 2, 4, 1, 8, 5, 1, 3, 1, 1, 3, 10, 1, 1, 1, 1, 4, 4

int compute_score(string word);

int main(void)
{
    // Prompt the user for two words
    string word1 = get_string("Player 1: ");
    string word2 = get_string("Player 2: ");

    // Compute the score of each word
    int score1 = compute_score(word1);
    int score2 = compute_score(word2);

    // Print the winner
}

int compute_score(string word)
{
    // Compute and return score for word
}
```

Para implementar `compute_score`, primeiro tente encontrar o valor em pontos de uma única letra em `word`.

- Lembre-se que para encontrar o caractere no índice n de uma string, `s`, você pode escrever `s[n]`. Então `word[0]`, por exemplo, irá lhe dar o primeiro caractere de `word`.
- Lembre-se que os computadores representam caracteres usando ASCII (http://asciitable.com/), um padrão que representa cada caractere como um número.
- Lembre-se também de que o índice 0 de `POINTS`, `POINTS[0]`, fornece o valor em ponto de 'A'. Pense em como você poderia transformar a representação numérica de 'A' no índice do seu valor em ponto. Agora, e quanto a 'a'? Você precisará aplicar transformações diferentes para letras maiúsculas e minúsculas, então as funções `isupper` (https://manual.cs50.io/3/isupper)e podem `islower` (https://manual.cs50.io/3/islower)ser úteis para você.
- Lembre-se de que caracteres que *não* sejam letras devem receber zero pontos. Por exemplo, `!` vale 0 pontos.

Se você conseguir calcular corretamente o valor de *um* caractere em `word`, provavelmente conseguirá usar um loop para somar os pontos dos demais caracteres. Depois de tentar o que foi descrito acima, considere esta dica (bastante reveladora!) abaixo.

```c
#include <ctype.h>
#include <cs50.h>
#include <stdio.h>
#include <string.h>

// Points assigned to each letter of the alphabet
int POINTS[] = {1, 3, 3, 2, 1, 4, 2, 4, 1, 8, 5, 1, 3, 1, 1, 3, 10, 1, 1, 1, 1, 4, 4

int compute_score(string word);

int main(void)
{
    // Prompt the user for two words
    string word1 = get_string("Player 1: ");
    string word2 = get_string("Player 2: ");

    // Compute the score of each word
    int score1 = compute_score(word1);
    int score2 = compute_score(word2);

    // Print the winner
}

int compute_score(string word)
{
    // Keep track of score
    int score = 0;

    // Compute score for each character
    for (int i = 0, len = strlen(word); i < len; i++)
    {
        if (isupper(word[i]))
        {
            score += POINTS[word[i] - 'A'];
        }
        else if (islower(word[i]))
        {
            score += POINTS[word[i] - 'a'];
        }
    }

    return score;
}
```

Finalmente, conclua a última etapa do seu pseudocódigo: imprimir o vencedor. Lembre-se de que uma `if` instrução `if` pode ser usada para verificar se uma condição é verdadeira e que o uso adicional de `else if` `or` `else` pode verificar outras condições (exclusivas).

```c
if (/* Player 1 wins */)
{
    // ...
}
else if (/* Player 2 wins */)
{
    // ...
}
```

```
else
{
    // ...
}
```

E depois de experimentar o que foi sugerido acima, fique à vontade para dar uma olhada na dica (ou melhor, na solução completa!) abaixo.

```
#include <ctype.h>
#include <cs50.h>
#include <stdio.h>
#include <string.h>

// Points assigned to each letter of the alphabet
int POINTS[] = {1, 3, 3, 2, 1, 4, 2, 4, 1, 8, 5, 1, 3, 1, 1, 3, 10, 1, 1, 1, 1, 4, 4

int compute_score(string word);

int main(void)
{
    // Prompt the user for two words
    string word1 = get_string("Player 1: ");
    string word2 = get_string("Player 2: ");

    // Compute the score of each word
    int score1 = compute_score(word1);
    int score2 = compute_score(word2);

    // Print the winner
    if (score1 > score2)
    {
        printf("Player 1 wins!\n");
    }
    else if (score1 < score2)
    {
        printf("Player 2 wins!\n");
    }
    else
    {
        printf("Tie!\n");
    }
}

int compute_score(string word)
{
    // Keep track of score
    int score = 0;

    // Compute score for each character
    for (int i = 0, len = strlen(word); i < len; i++)
    {
        if (isupper(word[i]))
        {
            score += POINTS[word[i] - 'A'];
        }
        else if (islower(word[i]))
        {
            score += POINTS[word[i] - 'a'];
```

```
        }
    }

    return score;
}
```

Como testar

Seu programa deve se comportar conforme os exemplos abaixo.

```
$ ./scrabble
Player 1: Question?
Player 2: Question!
Tie!
```

```
$ ./scrabble
Player 1: red
Player 2: wheelbarrow
Player 2 wins!
```

```
$ ./scrabble
Player 1: COMPUTER
Player 2: science
Player 1 wins!
```

```
$ ./scrabble
Player 1: Scrabble
Player 2: wiNNeR
Player 1 wins!
```

Correção

No seu terminal, execute o comando abaixo para verificar se o seu trabalho está correto.

```
check50 cs50/problems/2025/x/scrabble
```

Estilo

Execute o comando abaixo para avaliar o estilo do seu código usando `style50`.

```
style50 scrabble.c
```

Como enviar

No seu terminal, execute o comando abaixo para submeter seu trabalho.

```
submit50 cs50/problems/2025/x/scrabble
```