

Exercício: Horas

Harvard CS50 Staff

2023-09-02

Resumo

Este exercício corresponde ao problema *hours* 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	Horas de estudo	1
3	Detalhes de implementação	2
4	Como testar seu código?	2
5	Como enviar seu código?	3

1 Introdução

Este exercício corresponde ao problema “*hours*” 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 aprenda a utilizar *arrays*, saiba passar *arrays* como argumentos para funções, aprenda a adicionar valores utilizando um loop, pratique divisão inteira e modificação de tipo (*type casting*).

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/hours/)¹.

2 Horas de estudo

Suponha que você esteja fazendo CS50 (se você está lendo isso, provavelmente está!) e gastando tempo toda semana para estudar e fazer os PSETs. Você pode estar se perguntando quantas horas no total já gastou estudando ciência da computação, ou se perguntando quantas horas de estudo, em média, você estuda por semana.

Neste exercício você completará uma função que calcula, com base na entrada do usuário, o número total de horas estudadas ou a média de horas de estudo por semana. Algumas dicas:

- Para somar números em um array, você deve criar uma variável e inicializá-la com o valor zero. Depois você deve percorrer o array com um loop que adiciona cada valor do array à variável que você criou.
- Preste atenção no que ocorre se você dividir dois números inteiros (`int`) ao calcular a média.

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

3 Detalhes de implementação

Utilize o *starter file* com o nome “horas.c”. Nesse arquivo nós já implementamos toda a função `main` para você! Seu trabalho deve ser:

1. Estudar em detalhes a função `main` para aprender como ela funciona; e
2. Implementar a função `calcular_horas`, que é utilizada pela função `main`.

Note que a função `main` solicita ao usuário o número de semanas que ele está estudando a CS50 e, então, cria um *array* com esse número de elementos. Observe que, após coletar alguns dados, o programa pede ao usuário para digitar ‘T’ ou ‘M’ — ‘T’ deveria (mas ainda não o faz!) exibir o total de horas inseridas pelo usuário, enquanto ‘M’ deveria (mas ainda não o faz!) exibir a média de horas semanais de estudo do usuário.

Note também que o loop `do while` usa a função `toupper` para capitalizar a letra inserida antes de ser salva na variável `saida`. A função `toupper` é fornecida pela biblioteca (*header file*) `ctype.h`. Dê uma olhada na [documentação da biblioteca ctype.h](#) para entender como a função `toupper` funciona².

Por fim a função `main` chama a função `printf` que utiliza a função `calcular_horas` para mostrar o resultado correto.

Sintaxe com *arrays*

Preste muita atenção ao modo como os *arrays* são criados e especialmente em:

- Como informamos que um *array* é um parâmetro de uma função; e
- Como passamos um *array* como argumento para uma função.

A sintaxe para criar um *array*, usá-lo como parâmetro e passá-lo como argumento é semelhante, mas diferente. Certifique-se de que você entendeu isso!

O seu trabalho de programação é completar a função `calcular_horas`, escrevendo o código onde está o “TODO” dentro da função. Estude a documentação da função (os comentários logo acima da função) para saber exatamente quais os parâmetros e o retorno. A tarefa de programação não é difícil: primeiro some as horas salvas no *array* em uma nova variável. Então, dependendo do valor do argumento `saida`, retorne essa soma ou a média semanal de horas. Lembre-se de que você não deve alterar nada dentro da função `main`, mas deve entender como ela funciona para criar seu programa corretamente.

4 Como testar seu código?

O comportamento esperado do seu programa, se o usuário quiser ver a média semanal de horas de estudo, é o seguinte:

```
1 $ ./horas
2 Quantas semanas de estudo na CS50? 3
3 Horas na semana 0: 3
4 Horas na semana 1: 7
5 Horas na semana 2: 10
6 Digite T para o total ou M para a média de horas por semana: M
7 6.7 horas
8 $
```

²<https://manual.cs50.io/#ctype.h>

Já se o usuário quiser ver o total de horas estudadas, o comportamento esperado é o seguinte:

```
1  $ ./horas
2  Quantas semanas de estudo na CS50? 2
3  Horas na semana 0: 2
4  Horas na semana 1: 8
5  Digite T para o total ou M para a média de horas por semana: T
6  10.0 horas
7  $
```

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/)³.

5 Como enviar seu código?

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

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