

Este é o CS50x

OpenCourseWare

Doar [🔗](https://cs50.harvard.edu/donate) (<https://cs50.harvard.edu/donate>)

David J. Malan (<https://cs.harvard.edu/malan/>)

malan@harvard.edu

[f](https://www.facebook.com/dmalan) (<https://www.facebook.com/dmalan>) [G](https://github.com/dmalan) (<https://github.com/dmalan>) [@](https://www.instagram.com/davidjmalan/)

(<https://www.instagram.com/davidjmalan/>) [in](https://www.linkedin.com/in/malan/) (<https://www.linkedin.com/in/malan/>)

[ID](https://orcid.org/0000-0001-5338-2522) (<https://orcid.org/0000-0001-5338-2522>) [Q](https://www.quora.com/profile/David-J-Malan)

(<https://www.quora.com/profile/David-J-Malan>) [G](https://www.reddit.com/user/davidjmalan)

(<https://www.reddit.com/user/davidjmalan>) [T](https://twitter.com/davidjmalan) (<https://twitter.com/davidjmalan>)

Pluralidade

Implemente um programa que execute uma eleição de pluralidade, conforme a seguir.

```
$ ./plurality Alice Bob Charlie
Number of voters: 4
Vote: Alice
Vote: Bob
Vote: Charlie
Vote: Alice
Alice
```

Fundo

As eleições vêm em todas as formas e tamanhos. No Reino Unido, o **primeiro-ministro** (<https://www.parliament.uk/education/about-your-parliament/general-elections/>) é oficialmente nomeado pelo monarca, que geralmente escolhe o líder do partido político que ganha mais cadeiras na Câmara dos Comuns. Os Estados Unidos usam um processo de **Colégio Eleitoral** de (<https://www.archives.gov/federal-register/electoral-college/about.html>) várias etapas, no qual os cidadãos votam em como cada estado deve alocar os Eleitores que então elegem o Presidente.

Talvez a maneira mais simples de realizar uma eleição, no entanto, seja por meio de um método comumente conhecido como “voto de pluralidade” (também conhecido como “primeiro

após o post” ou “o vencedor leva tudo”). Na votação por pluralidade, cada eleitor pode votar em um candidato. No final da eleição, o candidato que obtiver o maior número de votos é declarado vencedor da eleição.

Começando

Veja como baixar o “código de distribuição” desse problema (ou seja, código inicial) em seu próprio CS50 IDE. Faça login no [CS50 IDE \(https://ide.cs50.io/\)](https://ide.cs50.io/) e, em uma janela de terminal, execute cada um dos [itens \(https://ide.cs50.io/\)](https://ide.cs50.io/) abaixo.

- Execute `cd ~` (ou simplesmente `cd` sem argumentos) para garantir que você está em seu diretório inicial).
- Execute `mkdir pset3` para fazer (ou seja, criar) um diretório chamado `pset3`.
- Execute `cd pset3` para mudar para (ou seja, abrir) esse diretório.
- Execute `mkdir plurality` para fazer (ou seja, criar) um diretório chamado `plurality` em seu `pset3` diretório.
- Execute `cd plurality` para mudar para (ou seja, abrir) esse diretório.
- Execute `wget https://cdn.cs50.net/2020/fall/psets/3/plurality/plurality.c` para baixar o código de distribuição deste problema.
- Execute `ls`. Você deve ver o código de distribuição deste problema, em um arquivo chamado `plurality.c`.

Entendimento

Vamos agora dar uma olhada `plurality.c` e ler o código de distribuição que foi fornecido a você.

A linha `#define MAX 9` é alguma sintaxe usada aqui para significar que `MAX` é uma constante (igual a `9`) que pode ser usada em todo o programa. Aqui, representa o número máximo de candidatos que uma eleição pode ter.

O arquivo então define um `struct` chamado `candidate`. Cada um `candidate` tem dois campos: um `string` chamado `name` representando o nome do candidato e um `int` chamado `votes` representando o número de votos que o candidato possui. Em seguida, o arquivo define um array global de `candidates`, onde cada elemento é ele próprio a `candidate`.

Agora, dê uma olhada na `main` própria função. Veja se consegue descobrir onde o programa define uma variável global que `candidate_count` representa o número de candidatos na eleição, copia os argumentos da linha de comando para o array `candidates` e pede ao usuário para digitar o número de eleitores. Em seguida, o programa permite que cada eleitor digite um voto (vê como?), Chamando a `vote` função de cada candidato votado. Por fim, `main` faz uma

chamada para a `print_winner` função para imprimir o vencedor (ou vencedores) da eleição.

Se você olhar mais abaixo no arquivo, no entanto, notará que as funções `vote` e `print_winner` foram deixadas em branco. Esta parte depende de você concluir!

Especificação

Concluir a implementação de `plurality.c` tal forma que o programa simule uma eleição por pluralidade de votos.

- Conclua a `vote` função.
 - `vote` leva um único argumento, um `string` chamado `name`, que representa o nome do candidato que foi votado.
 - Se `name` corresponder a um dos nomes dos candidatos na eleição, atualize o total de votos desse candidato para contabilizar a nova votação. A `vote` função, neste caso, deve retornar `true` para indicar uma votação bem-sucedida.
 - Se `name` não corresponder ao nome de nenhum dos candidatos na eleição, nenhum total de votos deve mudar, e a `vote` função deve retornar `false` para indicar uma cédula inválida.
 - Você pode presumir que dois candidatos não terão o mesmo nome.
- Conclua a `print_winner` função.
 - A função deve imprimir o nome do candidato que recebeu mais votos na eleição e, em seguida, imprimir uma nova linha.
 - É possível que a eleição termine em empate se vários candidatos tiverem, cada um, o número máximo de votos. Nesse caso, você deve imprimir os nomes de cada um dos candidatos vencedores, cada um em uma linha separada.

Você não deve modificar qualquer outra coisa no `plurality.c` outro do que as implementações dos `vote` e `print_winner` funções (e a inclusão de arquivos de cabeçalho adicionais, se quiser).

Uso

Seu programa deve se comportar de acordo com os exemplos abaixo.

```
$ ./plurality Alice Bob
Number of voters: 3
Vote: Alice
Vote: Bob
Vote: Alice
Alice
```

```
$ ./plurality Alice Bob  
Number of voters: 3  
Vote: Alice  
Vote: Charlie  
Invalid vote.  
Vote: Alice  
Alice
```

```
$ ./plurality Alice Bob Charlie  
Number of voters: 5  
Vote: Alice  
Vote: Charlie  
Vote: Bob  
Vote: Bob  
Vote: Alice  
Alice  
Bob
```

Passo a passo



Testando

Certifique-se de testar seu código para certificar-se de que ele lida com ...

- Uma eleição com qualquer número de candidatos (até o `MAX` de `9`)
- Votar em um candidato pelo nome
- Votos inválidos para candidatos que não estão na cédula
- Imprimir o vencedor da eleição se houver apenas um
- Imprimir o vencedor da eleição se houver vários vencedores

Execute o seguinte para avaliar a exatidão do seu código usando `check50`. Mas certifique-se de compilar e testar você mesmo!

```
check50 cs50/problems/2021/x/plurality
```

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

```
style50 plurality.c
```

Como enviar

Execute o procedimento a seguir, fazendo login com seu nome de usuário e senha do GitHub quando solicitado. Por segurança, você verá asteriscos (`*`) em vez dos caracteres reais em sua senha.

```
submit50 cs50/problems/2021/x/plurality
```