

Isso é CS50

Introdução do CS50 à Ciência da Computação

OpenCourseWare

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

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

malan@harvard.edu

 (<https://www.clubhouse.com/@davidjmalan>)  (<https://www.facebook.com/dmalan>) 

(<https://github.com/dmalan>)  (<https://www.instagram.com/davidjmalan/>) 

(<https://www.linkedin.com/in/malan/>)  (<https://orcid.org/0000-0001-5338-2522>) 

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

(<https://www.tiktok.com/@davidjmalan>)  (<https://davidjmalan.t.me/>) 

(<https://twitter.com/davidjmalan>)

Pluralidade

Para este programa, você implementará um programa que executa uma eleição de pluralidade, conforme abaixo.

```
$ ./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/) (<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 assentos na Câmara dos Comuns. [Os Estados Unidos usam um processo de Colégio Eleitoral](https://www.archives.gov/federal-register/electoral-college/about.html) (<https://www.archives.gov/federal-register/electoral-college/about.html>) de várias etapas, no qual os cidadãos votam sobre 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 posto” ou “o vencedor leva

tudo”). No voto de pluralidade, cada eleitor pode votar em um candidato. No final da eleição, o candidato que tiver o maior número de votos é declarado o vencedor da eleição.

Começando

Faça login em [code.cs50.io \(https://code.cs50.io/\)](https://code.cs50.io/), clique na janela do seu terminal e execute `cd` -o sozinho. Você deve descobrir que o prompt da janela do terminal é semelhante ao abaixo:

```
$
```

Próxima execução

```
wget https://cdn.cs50.net/2022/fall/psets/3/plurality.zip
```

para baixar um ZIP chamado `plurality.zip` em seu codespace.

Então execute

```
unzip plurality.zip
```

para criar uma pasta chamada `plurality`. Você não precisa mais do arquivo ZIP, então você pode executar

```
rm plurality.zip
```

e responda com “y” seguido de Enter no prompt para remover o arquivo ZIP que você baixou.

Agora digite

```
cd plurality
```

seguido de Enter para entrar (ou seja, abrir) nesse diretório. Seu prompt agora deve se parecer com o abaixo.

```
plurality/ $
```

Se tudo foi bem sucedido, você deve executar

```
ls
```

and see a file named `plurality.c`. Executing `code plurality.c` should open the file where you will type your code for this problem set. If not, retrace your steps and see if you can determine where you went wrong!

Understanding

Let's take a look at `plurality.c` and read through the distribution code that's been provided to you.

The line `#define MAX 9` is some syntax used here to mean that `MAX` is a constant (equal to `9`) that can be used throughout the program. Here, it represents the maximum number of candidates an election can have.

The file then defines a `struct` called a `candidate`. Each `candidate` has two fields: a `string` called `name` representing the candidate's name, and an `int` called `votes` representing the number of votes the candidate has. Next, the file defines a global array of `candidates`, where each element is itself a `candidate`.

Now, take a look at the `main` function itself. See if you can find where the program sets a global variable `candidate_count` representing the number of candidates in the election, copies command-line arguments into the array `candidates`, and asks the user to type in the number of voters. Then, the program lets every voter type in a vote (see how?), calling the `vote` function on each candidate voted for. Finally, `main` makes a call to the `print_winner` function to print out the winner (or winners) of the election.

If you look further down in the file, though, you'll notice that the `vote` and `print_winner` functions have been left blank. This part is up to you to complete!

Specification

Complete the implementation of `plurality.c` in such a way that the program simulates a plurality vote election.

- Complete the `vote` function.
 - `vote` takes a single argument, a `string` called `name`, representing the name of the candidate who was voted for.
 - If `name` matches one of the names of the candidates in the election, then update that candidate's vote total to account for the new vote. The `vote` function in this case should return `true` to indicate a successful ballot.
 - If `name` does not match the name of any of the candidates in the election, no vote totals should change, and the `vote` function should return `false` to indicate an invalid ballot.
 - You may assume that no two candidates will have the same name.
- Complete the `print_winner` function.
 - The function should print out the name of the candidate who received the most votes in the election, and then print a newline.
 - It is possible that the election could end in a tie if multiple candidates each have the maximum number of votes. In that case, you should output the names of each of the winning candidates, each on a separate line.

You should not modify anything else in `plurality.c` other than the implementations of the `vote` and `print_winner` functions (and the inclusion of additional header files, if you'd like).

Usage

Your program should behave per the examples below.

```
$ ./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
```

Walkthrough



Testing

Be sure to test your code to make sure it handles...

- An election with any number of candidate (up to the `MAX` of `9`)
- Voting for a candidate by name
- Invalid votes for candidates who are not on the ballot
- Printing the winner of the election if there is only one
- Printing the winner of the election if there are multiple winners

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

```
check50 cs50/problems/2023/x/plurality
```

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

```
style50 plurality.c
```

Como enviar

Em seu terminal, execute o abaixo para enviar seu trabalho.

```
submit50 cs50/problems/2023/x/plurality
```