

```

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

// Constante máximo de candidatos
#define MAX 9

// Estrutura candidatos, onde salva um nome (string) e a quantidade de
votos (int).
typedef struct
{
    string name;
    int votes;
}
candidate;

// Array de candidatos.
candidate candidates[MAX];

// Variável nº de candidatos, está sendo relatada fora da função main
para poder ser usada em todas as funções.
int candidate_count;

// Funções
bool vote(string name);
void print_winner(void);

int main(int argc, string argv[])
{
    // Checa se o usuário digitou menos de 1 candidato na execução do
    programa.
    if (argc < 2)
    {
        printf("Usage: plurality [candidate ...]\n");
        return 1;
    }

    // Conta o número de candidatos
    candidate_count = argc - 1;

    // Se o número de candidatos ultrapassa o MAX, retorna este erro.
    if (candidate_count > MAX)
    {
        printf("Maximum number of candidates is %i\n", MAX);
        return 2;
    }

    // Armazena o nome de cada candidato em uma variável candidates.
    for (int i = 0; i < candidate_count; i++)

```

```

{
    candidates[i].name = argv[i + 1];
    candidates[i].votes = 0;
}

// Solicita o número de eleitores.
int voter_count = get_int("Number of voters: ");

// Loop para digitar cada voto.
for (int i = 0; i < voter_count; i++)
{
    string name = get_string("Vote: ");

    // Função Vote.
    if (!vote(name))
    {
        printf("Invalid vote.\n");
    }
}

// Imprime os vencedores.
print_winner();
}

// Verifica se o nome digitado é válido e conta os votos.
bool vote(string name)
{
    // Loop onde compara-se o nome do candidato com o nome digitado no
    // voto, caso sejam iguais, soma-se 1 ao nº de votos do candidato.
    for(int i = 0; i < candidate_count; i++)
    {
        if(strcmp(candidates[i].name, name) == 0)
        {
            candidates[i].votes++;
            return true;
        }
    }
    return false;
}

// Imprime o(s) vencedor(es) da eleição
void print_winner(void)
{
    // Variável para armazenar os dígitos do bubble sort
    candidate temp[1];

    // loop para transportar o candidato com o maior nº de votos para o
    final do array

```

```

    for (int i = 0; i < candidate_count - 1; i++)
    {
        // Se o numero do array menor for maior que o numero do array
        // seguinte, inverte eles.
        // Se Candidate[1] = 3 e Candidate[2] = 1 então inverte e fica
        // Candidate[1] = 1 e Candidate[2] = 3.
        if(candidates[i].votes > candidates[i+1].votes)
        {
            temp[0] = candidates[i];
            candidates[i] = candidates[i + 1];
            candidates[i + 1] = temp[0];
        }
    }

    // Salva o valor de votos do candidato posicionado no último array.
    int maior = candidates[candidate_count - 1].votes;

    // Loop para passar candidato à candidato.
    for(int i = 0; i < candidate_count; i++)
    {
        // Se o numero de votos do candidato for = ao do candidato do
        // ultimo array, imprima ele como vencedor.
        if(candidates[i].votes == maior)
        {
            printf("%s\n", candidates[i].name);
        }
    }
    return;
}

```