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

// Max number of candidates
#define MAX 9

// Candidates have name and vote count
typedef struct
{
    string name;
    int votes;
}
candidate;

// Array of candidates
candidate candidates[MAX];

// Number of candidates
int candidate_count;

// Function prototypes
bool vote(string name);
void print_winner(void);

int main(int argc, string argv[])
{
    // Check for invalid usage
    if (argc < 2)
    {
        printf("Usage: plurality [candidate ...]\n");
        return 1;
    }

    // Populate array of candidates
    candidate_count = argc - 1;
    if (candidate_count > MAX)
    {
        printf("Maximum number of candidates is %i\n", MAX);
        return 2;
    }
    for (int i = 0; i < candidate_count; i++)
    {
        candidates[i].name = argv[i + 1];
        candidates[i].votes = 0;
    }

    int voter_count = get_int("Number of voters: ");

    // Loop over all voters
    for (int i = 0; i < voter_count; i++)
    {
        string name = get_string("Vote: ");

        // Check for invalid vote
        if (!vote(name))
        {
            printf("Invalid vote.\n");
        }
    }

    // Display winner of election
    print_winner();
}

// Update vote totals given a new vote
```

```
bool vote(string name)
{
    // TODO
    for (int i = 0; i < candidate_count ; i++)
    {
        if (strcmp(candidates[i].name, name) == 0)
        {
            // Update votes and return true
            candidates[i].votes++;
            return true;
        }
    }
    return false;;
}

// Print the winner (or winners) of the election
void print_winner(void)
{
    // TODO
    int largest_votes = 0;
    // Find out the largest number of votes
    for (int i = 0; i < candidate_count ; i++)
    {
        if (candidates[i].votes > largest_votes)
        {
            largest_votes = candidates[i].votes;
        }
    }
    // Print out the winner/winners
    for (int i = 0; i < candidate_count ; i++)
    {
        if (candidates[i].votes == largest_votes)
        {
            printf("%s\n", candidates[i].name);
        }
    }

    return;
}
```