

(a)

```
private int numFirstVotes(String candidate,
                           List<String> candidateList)
{
    int numVotes = 0;

    for (Ballot voterBallot : ballotList)
    {
        String first = voterBallot.firstChoiceFrom(candidateList);

        if (candidate.equals(first))
            numVotes++;
    }
    return numVotes;
}
```

(b)

```
public List<String> candidatesWithFewest(
    List<String> candidateList)
{
    int[] votes = new int[candidateList.size()];
    int minVotes = ballotList.size();

    for (int c = 0; c < candidateList.size(); c++)
    {
        String candidate = candidateList.get(c);
        votes[c] = numFirstVotes(candidate, candidateList);

        if (votes[c] < minVotes)
            minVotes = votes[c];
    }

    List<String> result = new ArrayList<String>();
    for (int c = 0; c < candidateList.size(); c++)
    {
        if (votes[c] == minVotes)
            result.add(candidateList.get(c));
    }

    return result;
}
```

(b) Alternate solution

```
public List<String> candidatesWithFewest(  
    List<String> candidateList)  
{  
    List<String> result = new ArrayList<String>();  
    int minVotes = ballotList.size() + 1;  
  
    for (int c = 0; c < candidateList.size();c++)  
    {  
        String candidate = candidateList.get(c);  
        int thisVotes = numFirstVotes(candidate, candidateList);  
  
        if (thisVotes < minVotes)  
        {  
            minVotes = thisVotes;  
            result = new ArrayList<String>();  
        }  
        if (thisVotes == minVotes)  
            result.add(candidateList.get(c));  
    }  
  
    return result;  
}
```