White Board

Step 1 Read the number of candidates and store it in an integer variable.

Step 2 Read the candidate name and party into a list using a for loop. Use one list for candidate name and another for party.

Step 3 Step 3 Create a list to track the final count of votes for each candidate (VoteCount).. This is the third list created.

Step 4 Read the number of votes cast and store it in an integer variable.

Step 5 Add the names from the votes cast to a list. This would be the fourth and final list needed.

Step 6 Loop through the list of Candidates and loop through the list of names on the votes cast to determine the VoteCount of each candidate

Step 7 Loop through the Candidates VoteCount to determine who the winner is with the highest votes. Also.. get the party name of the winning candidate.

Step 8 Check for a tie. Loop through the Candidates VoteCount to see if more than one candidate has the winning vote count.

Step 9 Display results. Either that there is a tie or return the party name of the winning candidate.

GitHub Link

<https://github.com/dleadbet/ConsoleAppElection02Lists>

C# Codes

using System;

using System.Collections.Generic;

namespace ConsoleAppElection02Lists

{

class Program

{

static void Main(string[] args)

{

List<string> candidateNameList = new List<string>();

List<string> candidatePartList = new List<string>();

List<int> candidateVotesList = new List<int>();

//Step 1 Read the number of candidates and store it in an integer variable.

int CandidateCount = int.Parse(Console.ReadLine());

//Step 2 Read the candidate name and party into a list using a for loop. Use one list for candidate name and another for party.

//Step 3 Create a list to track the final count of votes for each candidate. This is the third list created.

for (int i = 0; i < CandidateCount; i++)

{

candidateNameList.Add(Console.ReadLine());

candidatePartList.Add(Console.ReadLine());

candidateVotesList.Add(0); /\* Just initialize votes to zero \*/

}

//Step 4 Read the number of votes and store it in an integer variable.

int totalVoteCount = int.Parse(Console.ReadLine());

List<string> VotesList = new List<string>();

// Step 5 Add the names from the votes cast to a list. This would be the fourth and final list needed.

for (int i = 0; i < totalVoteCount; i++)

{

VotesList.Add(Console.ReadLine());

}

//Step 6 Loop through the list of Candidates and loop through the list of names on the votes cast to determine the VoteCount of each candidate

//Loop through each Candidate Name in the list

for (int i = 0; i < candidateNameList.Count; i++)

{

//Loop through each vote to count the votes for each candidate

for (int j = 0; j < VotesList.Count; j++)

{

if (candidateNameList[i] == VotesList[j])

{

candidateVotesList[i] += 1;

}

}

}

// Step 7 - Loop through the Candidates VoteCount to determine who the winner is with the highest votes. Also.. get the party name of the winning candidate.

//Find the index of the winnder in candidateList

int WinningCandidateIndex = 0;

// Find the int value of the most votes

int WinningCandidateVoteCount = 0;

for (int i = 0; i < candidateVotesList.Count; i++)

{

// find the highest vote count and the index of the Candidate

if (candidateVotesList[i] > WinningCandidateVoteCount)

{

WinningCandidateVoteCount = candidateVotesList[i];

WinningCandidateIndex = i;

}

}

//Step 8 - Check for a tie. Loop through the Candidates VoteCount to see if more than one candidate has the winning vote count.

//The WinningCandidateVoteCount is known so check if more than one candidate has that many votes

int tieCountCheck = 0;

for (int i = 0; i < candidateVotesList.Count; i++)

{

if (candidateVotesList[i] == WinningCandidateVoteCount)

{

tieCountCheck += 1;

}

}

if (tieCountCheck > 1)

{

Console.WriteLine("tie");

}

else

{

//If no tie then return the party of the winning candidate

Console.WriteLine(candidatePartList[WinningCandidateIndex]);

}

}

}

}

Kattis Screen Shot

