// Marissa Gross

// Project 3 - Pirate Election

// This program opens a file, gathers and analyzes the election votes,

// gets rid of the duplicate votes and determines the winner. It

// displays the election results and writes the legitimate votes to

// a new file called Counted.txt

#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

using namespace std;

//Constant variables

const int SIZE=100, YN=10;

//Function Prototypes

void sortData(ifstream &fileReader);

void createAmends(int row, char tempChars[SIZE], char ynVotes[][YN]);

void newCaptain(string vote, string newCapt[], int count);

void tallyResults(char ynVotes[][YN], string newCapt[], bool amendResults[], string &capResult, int invalid, int valid);

void createNewFile(int pID[], char vCh[][YN], string capV[], int votes);

int main()

{

ifstream fileReader;

fileReader.open("Ballots.txt");

if (fileReader.fail())

{

cout << "The ballots failed to open with code: " << fileReader.failbit;

fileReader.close();

}

else

sortData(fileReader);

fileReader.close();

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to analyze and gather the file data \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void sortData(ifstream &fileReader)

{

string newCapt[SIZE]; //array of votes for the new captain

char ynVotes[SIZE][YN]; //2D array of the amendment votes

char tempChars[YN]; //temporary array to hold YES/NO amendment votes

int ID[SIZE]={0}; //array to hold the pirates ID #

bool amendResults[YN] = {false};

int validCount = 0,

invalidCount = 0,

pirateID =0;

string vote = "",

invalid = "",

capResult = "";

//Read in the votes from Ballots.txt

while(!fileReader.eof())

{

fileReader >> pirateID;

if(!ID[pirateID-1])

{

ID[pirateID-1] = pirateID;

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

{

fileReader >> tempChars[i];

}

createAmends(validCount, tempChars, ynVotes);

getline(fileReader, vote);

newCaptain(vote, newCapt, validCount);

validCount++;

}

else

{

getline(fileReader, invalid);

invalidCount++;

}

}

tallyResults(ynVotes, newCapt, amendResults, capResult, invalidCount, validCount);

createNewFile(ID, ynVotes, newCapt, validCount);

cout << validCount << "\n";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to tally the votes and determine the winner of the new captain position, the new first \*

// mate and whether or not each amendment passed and to display all of the results \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void tallyResults(char ynVotes[][YN], string newCapt[], bool amendResults[], string &capResult, int invalid, int valid)

{

int totalVotes=invalid+valid;

int amRes[YN] = {0};

int cap = 0;

for(int r = 0;r<SIZE;r++)

{

for(int c = 0;c<YN;c++)

{

if (ynVotes[r][c] == 'Y')

{

amRes[c] = amRes[c]++;

}

}

}

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

{

if (amRes[i] > (.66\*valid))

{

amendResults[i] = true;

}

}

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

{

if(newCapt[i]=="The Captain")

{

cap++;

}

}

if(cap>(.50\*valid))

{

capResult = "The Captain";

}

else{

capResult = "The New Guy";

}

cout << " " << valid << " total pirates voted in this election using " << totalVotes << " ballots.\n\n";

if (capResult=="The Captain")

{

cout << " The new captain is \"The Captain.\" He received " << cap << " votes.\n";

cout << " And the new first mate is \"The New Guy.\" He received " << (valid-cap) << " votes.\n\n";

}

else

{

cout << " The new captain is \"The New Guy.\" He received " << (valid-cap) << " votes.\n";

cout << " And the new first mate is \"The Captain.\" He received " << cap << " votes.\n\n";

}

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

{

double temp=0;

if (amendResults[i]==1)

{

temp=amRes[i];

cout << " Amendment # " << (i+1) << " passes: (" << fixed << setprecision(0) << temp << " Y, " << valid-temp << " N) - ";

cout << fixed << setprecision(2) << ((temp/valid)\*100) << "% meets a 2/3 majority.\n";

}

else

{

temp=amRes[i];

cout << " Amendment # " << (i+1) << " fails: (" << fixed << setprecision(0) << temp << " Y, " << valid-temp << " N) - ";

cout << fixed << setprecision(2) << ((temp/valid)\*100) << "% is not a 2/3 majority.\n";

}

}

cout << "\n\n";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to analyze the string holding the pirates' votes \*

// for the new captain and creates an array of this data \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void newCaptain(string vote, string newCapt[], int count)

{

char t = vote[4];

if (t == 'C')

{

newCapt[count]="The Captain";

}

else if (t == 'N')

{

newCapt[count]="The New Guy";

}

else

return;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to create 2D array of the amendment votes \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void createAmends(int row, char tempChars[], char ynVotes[][YN])

{

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

{

ynVotes[row][i] = tempChars[i];

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function to create and write new counted.txt file \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void createNewFile(int pID[], char vCh[][YN], string capV[], int votes)

{

ofstream newFile;

newFile.open("Counted.txt");

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

{

if (pID[i]!=0)

{

newFile << pID[i];

for (int j=0; j<YN; j++)

{

newFile << vCh[i][j];

}

newFile << capV[i] << " \n";

}

}

newFile.close();

}