bool deleteKitten(string name, roster &kittenRoster){

if(findKitten(name, kittenRoster)==-1){

return false;

}

else{

for(int i=0; i<kittenRoster.size; i++)

{

if(kittenRoster.kittens[i].name==name)

{

for(int j=i; j<(kittenRoster.size-1); j++)

{

kittenRoster.kittens[j].name=kittenRoster.kittens[j+1].name;

kittenRoster.kittens[j].color=kittenRoster.kittens[j+1].color;

kittenRoster.kittens[j].score=kittenRoster.kittens[j+1].score;

}

break;

}

}

return true;

}

kittenRoster.size-=1;

}

#include <iostream>

#include <string>

using namespace std;

string name;

string color;

int score;

};

struct roster{

kitten kittens[10];

int size;

};

char printMenu(){

char playerOption;

do {

cout<<endl;

cout<<"MENU"<<endl;

cout<<"a - Add kitten"<<endl;

cout<<"d - Remove kitten"<<endl;

cout<<"u - Update kitten color and cuteness score"<<endl;

cout<<"r - Output kittens above a rating"<<endl;

cout<<"o - Output roster"<<endl;

cout<<"q - Quit"<<endl;

cout<<endl;

cout<<"Choose an option:"<<endl;

cin>>playerOption;

if(playerOption=='a'){

return 'a';

continue;

}

else if(playerOption=='d'){

return 'd';

continue;

}

else if(playerOption=='u'){

return 'u';

continue;

}

else if(playerOption=='r'){

return 'r';

continue;

}

else if(playerOption=='o'){

return 'o';

continue;

}

} while( playerOption!='q' );

return 0;

}

int findKitten(string name, roster kittenRoster){

int j=0;

for (j=0;j<kittenRoster.size;j++){

if (kittenRoster.kittens[j].name==name){

return j;

}

else{

return -1;

}

}

return 0;

}

roster addKitten(struct kitten, roster kittenRoster){

if(kittenRoster.size==10){

cout<<"Impossible to add new kitten: roster is full."<<endl;

}

else{

kittenRoster.kittens[kittenRoster.size+1]=kitten;

kittenRoster.size+=1;

cout<<"Successfully added new kitten to roster.";

}

return kittenRoster;

}

int main()

{

roster kittenRoster;

return 0;

}

bool deleteKitten(string name, roster kittenRoster){

if(findKitten(name)!='false'){

return false;

}

Else{

for (j=0;j<kittenRoster.size;j++){

kittenRoster.kittens[name];

}

kittenRoster.size-=1;

return true;

}

}

roster getKittenFromFile(test1, roster kittenRoster) {

ifstream testKittens;

string kittenName;

string kittenColor;

int kittenScore;

testKittens.open("test1.txt");

if (!testKittens.is\_open()) {

cout << "Error! File not found." << endl;

return 1;

}

while (!testKittens.fail()) {

getline(testKittens, kittenName);

getline(testKittens, kittenColor);

testKittens>> numWins;

kittenRoster.kittens[kittenRoster.size+1].name=kitten.name;

kittenRoster.kittens[kittenRoster.size+1].color=kitten.color;

kittenRoster.kittens[kittenRoster.size+1].score=kitten.score;

kittenRoster.size+=1;

teamFS.ignore();

teamFS.clear();

}

testKittens.close();

}

int getKittenFromFile(fstream& test1, roster kittenRoster) {

ifstream testKittens;

string kittenName;

string kittenColor;

int kittenScore;

testKittens.open("test1.txt");

if (!testKittens.is\_open()) {

cout << "Error! File not found." << endl;

return 1;

}

while (!testKittens.fail()) {

getline(testKittens, kittenName);

getline(testKittens, kittenColor);

testKittens>> numWins;

kittenRoster.kittens[kittenRoster.size+1].name=kitten.name;

kittenRoster.kittens[kittenRoster.size+1].color=kitten.color;

kittenRoster.kittens[kittenRoster.size+1].score=kitten.score;

kittenRoster.size+=1;

teamFS.ignore();

teamFS.clear();

}

testKittens.close();

}

bool updateKitten(struct kitten, roster kittenRoster){

if(findKitten(string kitten, roster kittenRoster)==-1){

Return false;

}

Else if{

kittenRoster.kittens[findKitten(string kitten.name, roster kittenRoster)]=kitten’

Return true;

}

}

printToFile(string filename, roster kittenRoster) {

cout<<"ROSTER"<<endl;

for (int j=0;j<kittens;j++){

cout<<"Kitten "<<j+1<<" -- Name: "<< kittenName[j] << ", Color: " << kittenColor[j] << ", Score: " <<kittenScore[j]<<endl;

}

}

printRoster(roster kittenRoster){

cout<<"ROSTER"<<endl;

for (int j=0;j<kittens;j++){

cout<<"Kitten "<<j+1<<" -- Name: "<< kittenName[j] << ", Color: " << kittenColor[j] << ", Score: " <<kittenScore[j]<<endl;

}

}