#include <iostream>

#include <sstream>

#include <string>

#include <fstream>

using namespace std;

struct kitten {

string name;

string color;

int score;

kitten():score(0){}

};

struct roster{

kitten kittens[10];

int size;

roster():size(0){}

};

char printMenu(){

cout<<"MENU"<<endl;

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

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

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

cout<<"f - Find kitten"<<endl;

cout<<"l - Load kitten info from file"<<endl;

cout<<"s - Save kitten info to file"<<endl;

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

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

cout<<endl;

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

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;

break;

}

else{

return -1;

break;

}

}

}

bool deleteKitten(string name, roster &kittenRoster){

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;

}

}

kittenRoster.size-=1;

return true;

}

bool updateKitten(kitten cat, roster &kittenRoster){

int catPos=findKitten(cat.name, kittenRoster);

kittenRoster.kittens[catPos].name=cat.name;

kittenRoster.kittens[catPos].color=cat.color;

kittenRoster.kittens[catPos].score=cat.score;

return true;

}

void printToFile(string fileName, roster kittens){

ofstream fileStream;

fileStream.open(fileName);

if(fileStream.is\_open())

{

cout << "It worked" << endl;

}

else{

cout << "Didnt work" << endl;

}

}

int getKittenFromFile(string fileName, 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>> kittenScore;

testKittens.ignore();

testKittens.clear();

}

testKittens.close();

return 0;

}

roster addKitten(kitten cat, roster &kittenRoster){

if(kittenRoster.size==10){

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

}

else{

kittenRoster.kittens[kittenRoster.size].name=cat.name;

kittenRoster.kittens[kittenRoster.size].color=cat.color;

kittenRoster.kittens[kittenRoster.size].score=cat.score;

kittenRoster.size+=1;

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

}

return kittenRoster;

}

int main()

{

roster kittenRoster;

char playerOption;

kitten newkitten;

string removeKitten;

string fKitten;

do {

printMenu();

cin>>playerOption;

if(playerOption=='a'){

cout<<"Enter a new kitten's name: "<<endl;

cin>>newkitten.name;

cout<<"Enter the kittens's color: "<<endl;

cin>>newkitten.color;

cout<<"Enter the kittens's cuteness score: "<<endl;

cin>>newkitten.score;

addKitten(newkitten, kittenRoster);

continue;

}

else if(playerOption=='d'){

if(deleteKitten(removeKitten, kittenRoster)){

cout<<"Enter kitten name to delete:"<<endl;

cin>>removeKitten;

}

else{

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

}

}

else if(playerOption=='u'){

cout<<"Enter a kitten name: "<<endl;

cin>>newkitten.name;

cout<<"Enter a new color for the kitten: "<<endl;

cin>>newkitten.color;

cout<<"Enter a new cuteness score for the kitten: "<<endl;

cin>>newkitten.score;

if(updateKitten(newkitten, kittenRoster))

{

cout<<"Successfully updated kitten info."<<endl;

cout<<endl;

}

else{

cout<<"Cannot find kitten."<<endl;

}

cout<<endl;

continue;

}

else if(playerOption=='f'){

cout<<"Enter a kitten name: "<<endl;

cin>>fKitten;

findKitten(fKitten, kittenRoster);

continue;

}

else if(playerOption=='l'){

continue;

}

else if(playerOption=='s'){

continue;

}

else if(playerOption=='o'){

continue;

}

} while( playerOption!='q' );

return 0;

}