#include <iostream>

#include <math.h>

#include <string.h>

using namespace std;

class FlightBooking {

public:

FlightBooking(int id, int capacity, int reserved);

FlightBooking();

void print();

void reserveSeats(int tiket);

void cancelReservations(int);

bool Checker();

int prectangle(int, int);

void CreateFlight(int id, int capacity);

void deleteFlight();

int get\_id();

private:

int id;

int capacity;

int reserved;

int reserve;

int ticket;

void set(int, int, int);

};

FlightBooking::FlightBooking() {

id = 0;

capacity = 0;

reserved = 0;

}

FlightBooking::FlightBooking(int id, int capacity, int reserved) {

set(id, capacity, reserved);

}

void FlightBooking::CreateFlight(int id, int capacity) { set(id, capacity, 0); }

void FlightBooking::deleteFlight() { set(0, 0, 0); }

void FlightBooking::set(int id, int capacity, int reserved) {

this->id = id;

this->capacity = capacity;

this->reserved = reserved;

}

int FlightBooking::prectangle(int capacity, int reserved) {

return (reserved \* 100) / capacity;

}

void FlightBooking::print() {

cout << "Flight " << this->id << ": " << this->reserved << "/" << this->capacity

<< " "

<< "(" << prectangle(capacity, reserved) << "%) seats reserved" << endl;

}

void FlightBooking::reserveSeats(int ticket) {

this->reserve = this->reserved + ticket;

if (!Checker()) {

if (106 > prectangle(this->capacity, this->reserve)) {

this->reserved += ticket;

goto Success;

}

}

cout << "-----------------------" << endl;

cout << "---Cannot perform this operation---" << endl;

cout << "-----------------------" << endl;

Success:

ticket = ticket;

}

void FlightBooking::cancelReservations(int Number) {

if (0 <= this->reserved - Number) {

this->reserved -= Number;

goto Success;

}

cout << "-----------------------" << endl;

cout << "---Cannot perform this operation---" << endl;

cout << "-----------------------" << endl;

Success:

Number = Number;

}

bool FlightBooking::Checker() {

if (prectangle(this->capacity, this->reserved) > 106) {

cout << "-----------------------" << endl;

cout << "---Cannot perform this operation---" << endl;

cout << "-----------------------" << endl;

return true;

}

if (this->capacity < 0 || this->reserved < 0) {

cout << "-----------------------" << endl;

cout << "---Cannot perform this operation---" << endl;

cout << "-----------------------" << endl;

return true;

}

if (this->id < 1) {

cout << "-----------------------" << endl;

cout << "---Incorrect ID---" << endl;

cout << "-----------------------" << endl;

return true;

}

return false;

}

int FlightBooking::get\_id() { return id; }

int main() {

int reserved;

int capacity;

int IDargument;

int argument;

int number;

int IDrecord;

int j, i;

string temp;

string command = "";

FlightBooking booking[10];

booking[0] = FlightBooking(1, 400, 0);

while (command != "quit") {

argument = 0;

IDargument = 0;

j = 0, i = 0;

cout << "What would you like to do?: ";

getline(cin, command);

if (command[0] == 'c' && command[1] == 'r') {

for (i = command.length() - 1; command[i] != ' '; i--, j++) {

temp = command[i];

number = stoi(temp);

argument += number \* pow(10, j);

}

for (i--, j = 0; command[i] != ' '; i--, j++) {

temp = command[i];

number = stoi(temp);

IDargument += number \* pow(10, j);

}

for (i = 0; i < 10; i++) {

if (booking[i].get\_id() == 0) {

if (IDargument >= 1) {

if (argument >= 0) {

booking[i].CreateFlight(IDargument, argument);

break;

}

} else {

cout << "-----------------------" << endl;

cout << "---Incorrect ID---" << endl;

cout << "-----------------------" << endl;

break;

}

}

if (i == 9) {

cout << "-----------------------" << endl;

cout << "---Maximum flights number---" << endl;

cout << "-----------------------" << endl;

}

}

} else if (command[0] == 'd') {

for (i = command.length() - 1; command[i] != ' '; i--, j++) {

temp = command[i];

number = stoi(temp);

IDargument += number \* pow(10, j);

}

for (i = 0; i < 10; i++) {

if (IDargument == booking[i].get\_id()) {

if (!booking[i].Checker()) {

booking[i].deleteFlight();

}

break;

}

if (i == 9) {

cout << "-----------------------" << endl;

cout << "---No flight with ID " << IDargument << " ---" << endl;

cout << "-----------------------" << endl;

}

}

} else if (command[0] == 'l') {

cout << "-----------------------" << endl

<< "List: " << endl

<< "-----------------------" << endl;

for (i = 0; i < 10; i++) {

if (booking[i].get\_id() != 0) {

booking[i].print();

}

}

cout << "-----------------------" << endl;

} else if (command[0] == 'a') {

for (i = command.length() - 1; command[i] != ' '; i--, j++) {

temp = command[i];

number = stoi(temp);

argument += number \* pow(10, j);

}

for (i--, j = 0; command[i] != ' '; i--, j++) {

temp = command[i];

number = stoi(temp);

IDargument += number \* pow(10, j);

}

for (i = 0; i < 10; i++) {

if (IDargument == booking[i].get\_id()) {

booking[i].reserveSeats(argument);

break;

}

if (i == 10) {

cout << "-----------------------" << endl;

cout << "---No flight with ID " << IDargument << " ---" << endl;

cout << "-----------------------" << endl;

}

}

} else if (command[0] == 'c' && command[1] == 'a') {

for (i = command.length() - 1; command[i] != ' '; i--, j++) {

temp = command[i];

number = stoi(temp);

argument += number \* pow(10, j);

}

for (i--, j = 0; command[i] != ' '; i--, j++) {

temp = command[i];

number = stoi(temp);

IDargument += number \* pow(10, j);

}

for (i = 0; i < 10; i++) {

if (IDargument == booking[i].get\_id()) {

booking[i].cancelReservations(argument);

break;

}

if (i == 9) {

cout << "-----------------------" << endl;

cout << "---No flight with ID " << IDargument << " ---" << endl;

cout << "-----------------------" << endl;

}

}

}

}

}