

## Pràctica 2

### Format i data de lliurament

Cal lliurar la Pràctica abans del dia **10 de juny a les 23:59h**. Per al lliurament caldrà que entregueu un fitxer en format **ZIP** de nom ***logincampus\_pr2*** en minúscules (on *logincampus* és el nom d'usuari amb el què feu login al Campus). El ZIP ha de contenir:

- Workspace CodeLite sencer, amb tots els fitxers que es demanen.

Per reduir la mida dels fitxers i evitar problemes d'enviament que es poden donar en incloure executables, cal eliminar el que genera el compilador. Podeu utilitzar l'opció "Clean" del workspace o eliminar-los directament (les subcarpetes Menu i Test són les que contenen tots els binaris que genera el compilador).

Cal fer el lliurament a l'apartat de lliuraments d'AC de l'aula de teoria.

### Presentació

Aquesta pràctica culmina el projecte començat a la pràctica 1. Hem construït una aplicació per gestionar els avions i els passatgers d'una empresa de transport aeri i ara ens demanen de gestionar els vols. Treballarem amb un TAD cua per representar les cues d'embarcament que permetran als diferents passatgers accedir als avions que els han de dur als seus destins.

### Competències

#### Transversals

- Capacitat de comunicació en llengua estrangera.

#### Específiques

- Capacitat de dissenyar i construir aplicacions informàtiques mitjançant tècniques de desenvolupament, integració i reutilització.
- Coneixements bàsics sobre l'ús i la programació dels ordinadors, sistemes operatius, i programes informàtics amb aplicació a l'enginyeria.

## Objectius

- Analitzar un enunciat i extreure'n els requeriments tant de tipus de dades com funcionals (algorismes)
- Analitzar, entendre i modificar adequadament codi existent
- Saber utilitzar tipus de dades abstractes
- Saber utilitzar punters

## Recursos

Per realitzar aquesta activitat teniu a la vostra disposició els següents recursos:

- Materials en format web de l'assignatura
- Laboratori de C

## Criteris de valoració

Cada exercici té associada la puntuació percentual sobre el total de l'activitat. Es valorarà tant la correctesa de les respostes com la seva completeness.

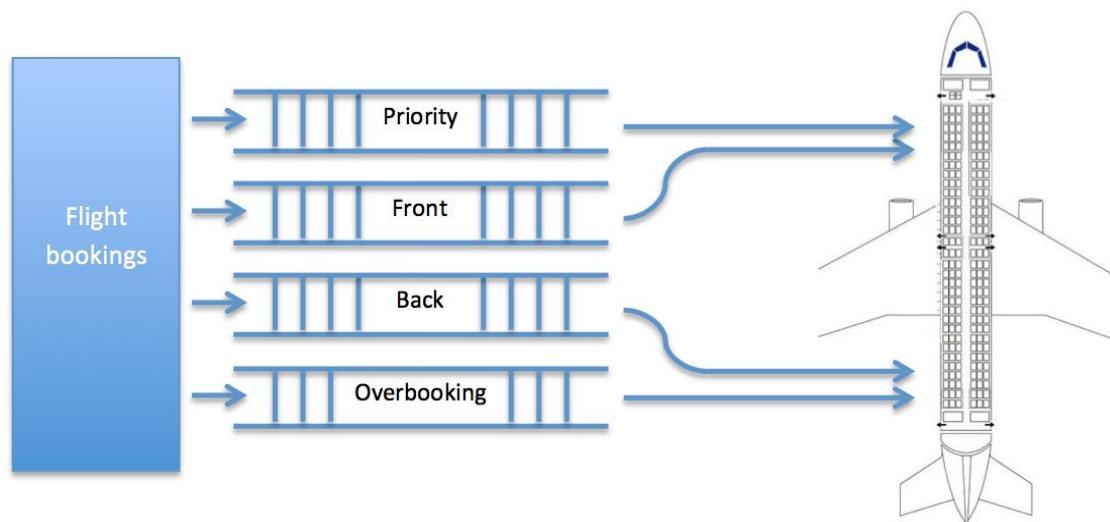
- Els exercicis en llenguatge C, han de compilarse per ser evaluats. En tal cas, es valorarà:
  - Que funcionin correctament
  - Que es respectin els criteris d'estil (Vegeu la Guia d'estil de programació en C que teniu a la Wiki)
  - Que el codi estigui comentat (preferiblement en anglès)
  - Que les estructures utilitzades siguin les adequades

## Descripció del projecte

En aquesta pràctica gestionarem l'embarcament de passatgers en l'avió que els ha de dur al seu destí. Disposarem d'una llista de passatgers que han fet la reserva a l'avió i diverses cues d'embarcament. En el cas de UOCAirways, s'organitzen les cues d'embarcament d'aquesta manera:

- Els usuaris que han comprat un seient preferent tenen assignat un seient de les primeres files de l'avió. Seran els que tinguin un seient en el primer 25% de files del avió. Se'ls donarà preferència per entrar a l'avió, per la part davantera.
- La resta de passatgers que no tenen seient preferent embarcaran per la porta del davant o del darrera de l'avió, atenent al seu número de seient. Les files que ocupen el següent 25% de l'avió embarquen per davant també, mentre que les files que ocupen el darrer 50% de l'avió embarquen per la porta del darrera. Després dels passatgers preferents, es dona pas simultani a aquests dos conjunts de passatgers perquè ocupen posicions diferents de l'avió i no es produirà cap col·lapse pel fet d'embarcar alhora.
- Per últim, a un passatger que no té seient assignat per embarcar se li dona una tarja d'embarcament provisional sense fila ni seient assignats. D'aquesta forma, es pot dirigir a la porta d'embarcament i esperar que entri tothom. Si, un cop es tanca l'embarcament, hi ha encara seients lliures, els passatgers d'*overbooking* poden entrar a l'avió, per la part posterior, en l'ordre en què han recollit la seva tarja provisional.

Esquemàticament:



Per exemple, amb un avió de 40 fileres:

- Files 1 a 10 (25%). Seients preferents. Embarquen per la porta del davant.
- Files 11 a 20 (25%). Seients normals. Embarquen per la porta del davant.
- Files 21 a 40 (50%). Seients normals. Embarquen per la porta del darrera.

Observeu que:

- Els passatgers entren a la llista de reserves (“*Flight bookings*”) en el moment que compren el vol per la web de la companyia.
- Quan s’obre el període de *check-in*, els passatgers obtenen la seva tarja d’embarcament, moment en què obtenen el número de seient dins l’avió (si encara hi ha seients disponibles).
- Amb la seva tarja d’embarcament i, en funció del número de fila assignada, s’han de dirigir a la cua d’embarcament corresponent, des de la qual accediran a l’avió quan els toqui el torn i sempre en l’ordre d’arribada a la cua.

Juntament amb l’enunciat se us facilita un nou workspace Codelite basat en la solució de la pràctica 1, però amb alguns canvis i evolucions. Aquest workspace serà la base per aquesta segona pràctica.

Veureu que han aparegut nous fitxers: **flight.c i .h** (que recull les operacions sobre els vols), **list.c i .h** (que recull les operacions sobre el TAD llista) i, per últim, **queue.c i .h** (que recull les operacions sobre el TAD cua). A més, s’han introduït els nous tipus de dades relacionats a **data.h**:

- **tBooking**, que representa una reserva d’un passatger en un vol, que pot ser en qualitat d’embarcament prioritari o no.
- **tBookingList**, per representar la llista de reserves que pot tenir un vol.
- **tBoarding**, que representa una tarja d’embarcament per la qual un passatger té assignat una fila i un seient individual de l’avió.
- **tBoardingQueue**, que representa una cua d’embarcament. Aquesta cua estarà composada per diversos elements tBoarding (passatgers amb tarja d’embarcament).
- **tFlight**, que representa un vol de l’aerolínia. Cada vol disposarà d’un identificador de vol, un avió associat, una porta d’embarcament, un origen, un destí, una data i una hora. A més, tindrà també la llista de reserves fetes pels passatgers i les quatre cues descrites anteriorment en l’enunciat.
- **tFlightTable**, que conté diversos elements de tipus tFlight. És la relació de vols que l’aerolínia està gestionant en un moment donat.
- El tipus **tPlane** incorpora també una taula bidimensional que representa la distribució i ocupació de seients dins l’avió (camps *layout* i *occupiedSeats*).
- També s’incorporen altres tipus auxiliars dels anteriors com són: **tFlightId** (identificador de vol), **tTime** (hora del vol), **tAirport** (codi de l’aeroport origen i destí), **tFlightStatus** (estat d’un vol).

A banda del codi font afegit i/o modificat, veureu que apareix també un nou fitxer de dades: el fitxer de vols (*flights.txt*). Aquest fitxer conté la persistència dels objectes de tipus *tFlight*, ordenats de menor a major data.

## Tests

El projecte CodeLite es presenta, com a la Pràctica 1, amb dos modes d'execució. No obstant, el mode Menu no incorpora cap entrada nova que us permeti provar interactivament la gestió dels vols. Això és així perquè es pretén que executeu i proveu el vostre programa en mode Test.

En mode Test podreu anar verificant el funcionament del vostre codi a mida que completeu els exercicis. Tingueu en compte que el resultat pot ser correcte encara que el codi no estigui bé (és a dir, el resultat no us dona una correcció definitiva) i que el total de tests superats no equival a la nota de la pràctica ja que el nombre de tests per apartat no es correspon amb el seu pes en la qualificació.

## Enunciat

### [25%] Exercici 1: Comprovació de dades

El processament dels vols s'inicia en l'acció `processAllFlights` (api.c). Des d'aquesta acció es van iterant els diferents vols i es van processant un a un, cridant a l'acció `processFlight` (api.c).

Si veieu, a `processFlight`, les diferents accions de què consta el processament d'un vol, comprobareu que el primer pas és comprovar que les dades del vol, passatgers i avions són correctes. Això es fa en l'acció `checkData`, que és la que es demana implementar en aquest exercici.

En concret, es demana que feu les següents comprovacions en aquest ordre:

- Comprovar que l'avió del vol existeix a la taula d'avions i que, en cas que existeixi, el seu nombre de seients sigui un múltiple de `MAX_SEATS_PER_ROW`. En cas que l'avió no existeixi es retornarà un error `ERR_NO_PLANE` com a valor de retorn a `RetVal`. En cas que no sigui múltiple, l'error retornat serà `ERR_INVALID_DATA`.
- Comprovar que tots els passatgers de la llista de reserves del vol (`bookings`) són passatgers registrats en la taula de passatgers que es rep per paràmetre. En cas que no sigui així, s'haurà de retornar un valor `ERR_NO_PASSENGER` a `RetVal`.
- Comprovar que el vol no està iniciat. El vol es considerarà iniciat si es dona alguna d'aquestes condicions: té un estat diferent de `BOOKING`, té una porta d'embarcament ja assignada o si alguna de les cues d'embarcament té ja algun passatger. L'error que es retornarà en cas que es detecti un vol iniciat serà `ERR_STARTED_FLIGHT`.

En cas que no es detecti cap situació errònia de les anteriors, es retornarà a `RetVal` el valor `OK`.

#### NOTES:

- Per saber si un avió existeix el podeu buscar a partir del seu identificador amb l'ajuda de la funció `planesTable_find`.
- Teniu disponibles operacions per treballar amb llistes de reserves a `list.c`.
- Per saber si un passatger existeix el podeu buscar a partir del seu identificador amb l'ajuda de la funció `passengerTable_find`.

## [25%] Exercici 2: Check-in del vol

Continuem analitzant el processament d'un vol individual (acció *processFlight*). Després de la comprovació de dades duta a terme en l'exercici anterior, si tot és correcte, es localitza l'avió que haurà de fer el trajecte i s'assigna una porta d'embarcament per al vol.

Un cop fet això, s'inicia el check-in pròpiament dit, mitjançant la crida a l'acció **checkInFlight** (api.c), que és l'acció que es demana implementar en aquest exercici.

En concret, es demana:

- Canviar l'estat del vol a CHECKIN.
- Recórrer la llista de reserves (*bookings*), mitjançant les operacions que ofereix el TAD Llista per fer-ho.
- Per cada reserva (*tBooking*), cal obtenir la seva corresponent tarja d'embarcament (*tBoarding*). Per fer-ho, cal invocar l'acció *assignSeat* (que ja es dona implementada i que només cal cridar).
- Amb la tarja d'embarcament obtinguda, ja només cal dirigir el passatger cap a la cua d'embarcament que correspongui. Caldrà comprovar el seient assignat al passatger i encuar-lo en la cua que li correspongui (*priority*, *front*, *back* o *overbooking*) segons la lògica detallada en l'enunciat general.

### NOTES:

- Per poder implementar la lògica de decisió que envia un passatger a la seva corresponent cua, caldrà conèixer els llindars que permeten saber fins a quina fila arriba la zona preferent, la que delimita la zona *front* i la que delimita la zona *back*. Teniu a la vostra disposició l'acció *getThresholdRows* per obtenir aquesta informació.
- L'acció *assignSeat* requereix dur el compte dels diferents seients de prioritat, *front* i *back* que es van assignant (ja que aquesta acció intenta anar assignant seients espaiadament en l'avió per repartir-ne el pes en la mesura del possible). Per tant, caldrà que declareu i inicialitzeu enters per dur aquest compte, i passar-los a la invocació a *assignSeat* convenientment.

## [25%] Exercici 3: Embarcament de passatgers

Seguint el fil d'anàlisi del processament d'un vol individual (acció *processFlight*), veiem que després del checkin es realitza una preparació de l'avió que ha de començar a admetre passatgers (*setUpPlane*).

Un cop l'avió està llist, s'inicia l'embarcament mitjançant la crida a **boardFlight** (api.c), que és l'acció que es demana implementar en aquest exercici.

En concret, es demana:

- Canviar l'estat del vol a BOARDING.
- Anar desencuant la cua de prioritat i acomodant els passatgers en l'avió. Per fer-ho, compteu amb l'acció *accommodatePassenger*.
- A continuació, anar desencuant simultàniament les cues de passatgers que han d'embarcar per la part davantera i posterior de l'avió. Això vol dir que s'haurà de donar pas alternatiu als passatgers d'una i altra cua (invocant *accommodatePassenger*).
- Com a darrer pas, s'haurà de desencuar la cua d'*overbooking* i intentar acomodar els passatgers en aquesta situació dins l'avió (invocant per cadascun d'ells l'acció *accommodatePassenger*).
- Sempre que s'aconsegueixi acomodar un passatger a l'avió, se li acumularan 100 punts al passatger en la seva targeta de fidelització (només en cas que en disposi d'una).

### NOTES:

- L'acció *accommodatePassenger* retorna OK si s'ha pogut acomodar el passatger en l'avió i un valor ERR\_NO\_FREE\_GAP en cas que no es pugui (cosa que no hauria de produir-se tret que hi hagi un cas d'*overbooking*). Si un passatger no es pot acomodar a l'avió, quedarà en la mateixa cua on es troava esperant.
- Per localitzar un passatger en la taula de passatgers a partir del seu identificador, podeu fer servir la funció *passengerTable\_find* (passenger.h). Aquesta funció us pot resultar útil en el moment que hagueu d'actualitzar els punts de fidelització del passatger.
- Teniu disponibles operacions per treballar amb cues de reserves a queue.c.

## [25%] Exercici 4: Tractament de l'overbooking

Tornem a l'acció que processa tots els vols de la companyia (*processAllFlights*).

El processament d'un vol individual finalitza quan l'embarcament dels passatgers ha acabat i el vol queda en estat CLOSED. Però, tot i que un vol quedí tancat, això no vol dir que tots els seus passatgers hagin pogut embarcar. N'hi poden haver alguns que no hagin pogut obtenir seient i que calgui reubicar en propers vols.

Es demana:

- a) **[10%]** Fer una cerca del proper vol que hagi de fer el mateix trajecte per poder reubicar les persones que s'han quedat sense volar. Per això, implementeu la funció **searchNextFlight** (api.c). Aquesta funció ha de buscar els vols que encara no s'han processat i trobar-ne un que faci el mateix trajecte (és a dir, amb els mateixos aeroports d'origen i destí).
- b) **[15%]** En cas que es trobi un vol amb aquestes característiques, es procedirà a reubicar els passatgers. Per fer-ho, caldrà que implementeu l'acció **reallocatesPassengers** (api.c), que haurà de moure tots els passatgers que hagin quedat en la cua d'*overbooking* a la llista de reserves del vol trobat en l'apartat anterior. Aquests passatgers s'inseriran al principi de la llista de reserves (*bookings*) en el mateix ordre que tenien en la cua, per assegurar que tindran un lloc assegurat en el proper vol. Per tant, passaran per davant dels passatgers que ja estiguessin en la llista de reserves del nou vol. Si són passatgers posseïdors de la tarja de fidelització de la companyia, seran inserits com a reserva amb prioritat. Si no ho són, seran inserits com a reserva sense prioritat.

A més, els passatgers que posseeixin la tarja de fidelització de la companyia, rebran una compensació en forma de 200 punts a la seva tarja en el moment en què siguin inserits en la llista de reserves del proper vol.

### NOTA:

- Per localitzar un passatger en la taula de passatgers a partir del seu identificador, podeu fer servir la funció *passengerTable\_find* (passenger.h). Aquesta funció us pot resultar útil en el moment que hagueu d'actualitzar els punts de fidelització del passatger.