Algoritmos y Estructuras de Datos II

Parcial 28-04: Tema A - Escalas en Aeropuerto

Ejercicio 1: Implementación de Escala de Vuelos

En el directorio del ejercicio se encuentran los siguientes archivos:

Archivo	***Descripción***
`main.c`	Contiene la función principal del programa
`flight.h`	Declaraciones relativas a la estructura de los datos de vuelos y de funciones de carga y escritura de datos.
`flight.c`	Implementaciones **incompletas** de las funciones
`array_helpers.h`	Declaraciones / prototipos de las funciones que manejan la tabla de escalas
`array_helpers.c`	Implementaciones **incompletas** de las funciones que manejan el arreglo

Abrir el archivo **`input/example10.in`** para ver cómo se estructuran los datos.

Cada línea contiene los datos de llegada (*arriival*) y partida (*departure*) de un vuelo particular. El primer dato corresponde al código de vuelo. Luego siguen **dos grupos de tres columnas** (seis columnas en total), las tres columnas de cada grupo se corresponden al *tipo de vuelo*, *hora* y *número de pasajeros*. El primer grupo representa la llegada para ese vuelo, y el segundo grupo la salida. El esquema es el siguiente:

<código>	<tipo-vuelo>	<hora-llegada>	<pasajeros>	<tipo-vuelo>	<hora-salida>	<pasajeros>

Consideraciones:

- La primera terna de datos siempre tiene tipo “0”: llegada
- La siguiente terna de datos siempre tiene tipo “1”: partida
- A lo largo del día, solo hay una llegada y una salida por hora.
- Un vuelo siempre sale en un horario posterior (o al menos igual) al horario en que llegó.
- Las horas siempre estarán en el rango 1 a 24 en los archivos de entrada
- Si se lee en el archivo una hora `h` debe interpretarse como la hora `h - 1` (ej: <7> ==> 6am).

El ejercicio consiste en completar el procedimiento de carga de datos en los archivos **`array_helpers.c`** y **`flight.c`**. Recordar que el programa tiene que ser *robusto*, es decir, debe tener un comportamiento bien definido para los casos en que la entrada no tenga el formato esperado.

Una vez completada la lectura de datos se puede verificar si la carga funciona compilando,

```
$ gcc -Wall -Werror -Wextra -pedantic -std=c99 -c array_helpers.c flight.c main.c
$ gcc -Wall -Werror -Wextra -pedantic -std=c99 array_helpers.o flight.o main.o -o layover
```

y luego ejecutar

```
$ ./layover input/example10.in
```

Ejercicio 2: Análisis de los datos

Completar la siguiente función, definida en **`array_helpers`**

```
unsigned int passengers_amount_in_airport(LayoverTable a, unsigned int hour);
```

Esta función debe retornar la cantidad de pasajeros que están esperando un vuelo en el aeropuerto en la hora ***hour***. Tener en cuenta que el resultado va a depender de lo que sucedió en las horas anteriores, y además que:

- Quienes arribaron en el vuelo de llegada de la hora ***hour*** se cuentan como pasajeros en espera, y se considera que el vuelo de salida de ese horario todavía no ocurrió.

Finalmente modificar el archivo **`main.c`** para que se muestre la cantidad de pasajeros en espera a las `10am`. Cada archivo de ejemplo incluye en el nombre la cantidad esperada.