

# Algoritmos y Estructuras de Datos II

## Parcial 3-05: Tema C - Entrega de Envíos

### Ejercicio 1: Implementación de Entrega de Envíos

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 <b>incompletas</b> de las funciones
array_helpers.h	Declaraciones / prototipos de las funciones que manejan la tabla de entregas
array_helpers.c	Implementaciones <b>incompletas</b> de las funciones que manejan el arreglo

Abrir el archivo `input/example1722.in` para ver cómo se estructuran los datos.

Cada línea contiene los datos de entrega de cajas (*boxes*) y cartas (*letters*) de un vuelo particular. El primer dato corresponde al código de vuelo y el segundo a la hora de llegada. Luego siguen **dos pares de columnas** donde cada par se corresponde al tipo de items y numero de items a entregar al deposito.

<código>	<hora-llegada>	<tipo>	<número-items>	<tipo>	<número-items>
----------	----------------	--------	----------------	--------	----------------

Consideraciones:

- El primer par de datos siempre se corresponde con el ítem tipo "0": boxes
- El siguiente par de datos siempre se corresponde con el ítem tipo "1": letters
- A lo largo del día, sólo se hace una entrega de cada tipo por hora.
- Las horas siempre estarán en el rango 1 a 24 en los archivos de entrada

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 delivery
```

y luego ejecutar

```
$ ./delivery input/example1722.in
```

## Ejercicio 2: Análisis de los datos

Completar la siguiente función, definida en `array_helpers`

```
unsigned int extra_space_fee_cost(DeliveryTable a, unsigned int hour);
```

Esta función debe retornar el costo que la empresa deberá pagar al depósito por la cantidad de ítems extra que guarda hasta la hora *hour*.

- Se paga un costo extra por cada ítem que supera el `MAX_ALLOWED` para ese día (hay una constante para cada tipo de ítem)
- Por cada ítem que supera el valor máximo permitido, se suma al costo el valor definido de `PENALTY`

Finalmente modificar el archivo `main.c` para que se muestre la cantidad de costo extra hasta las 8pm inclusive (los ítems que llegan en ese horario deben contarse). Cada archivo de ejemplo incluye en el nombre la cantidad esperada.