

Algoritmos y Estructuras de Datos II

Parcial 28-04: Tema B - Entrega de Envíos

Ejercicio 1: Implementación de Entrega de Envíos

En el directorio del ejercicio se encuentran los siguientes archivos:

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

Abrir el archivo `input/example1230.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. Los **siguientes dos pares de dos columnas cada uno** corresponden a la hora de llegada y numero de items a entregar al deposito. El esquema sería el siguiente:

<código>	<hora-llegada>	<número-items>	<hora-llegada>	<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, solo se hace una entrega de cada tipo por hora.
- 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: <19> ==> 6pm).

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/example1230.in
```

Ejercicio 2: Análisis de los datos

Completar la siguiente función, definida en `array_helpers`

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

Esta función debe retornar el costo que la empresa deberá pagar al depósito por la cantidad de items extra que guarda ese día.

- Los ítems para un día se calculan solo hasta las 6 de la tarde.
- 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 para el día (recordar que esto se calcula con los vuelos que ocurren hasta las 6pm incluido). Cada archivo de ejemplo incluye en el nombre la cantidad esperada.