

SINTAXIS Y SEMANTICA DE LENGUAJES
EXAMEN LENGUAJE C – 08/09/2016

NOTA _____

Nombre y Apellido _____ Com. _____

Prof.Teoría _____ Prof.Práctica _____

Una empresa de turismo instaló un nuevo sistema de ticket y necesita probar su eficiencia, para ello debemos analizar 200 ticket (como máximo) recibidos en los mails en el nuevo sistema.

De cada uno de los mails validos se almacenará: el cuerpo del mail y a que módulo (Hotel, Paquete, Vuelo) del sistema turístico pertenece.

Para almacenarlos utilizaremos la siguiente estructura:

```
struct tickCol {  
    char incid[200];  
    char modulo[15];  
};
```



* El sistema debe identificar a que módulo pertenece cada cuerpo del mensaje, para esto sabemos cómo regla que empieza con el formato “**módulo:**” (ej, “**Hotel:, Vuelo:, Paquete:**”). Ver imagen.

* Hotel, Paquete y Vuelo son los únicos módulos posibles.

* Si no se puede identificar a que módulo pertenece, el mail no se contabiliza en el sistema.

Como estadística el sistema debe indicar qué módulo tiene mayor ocurrencia. La salida se muestra en el main().

Ejemplo de salida: **El módulo de mayor ocurrencia es: Hotel, tiene 4 incidencias.**

Utilizar las siguientes funciones:

➤ **void carga_ticket (struct tickCol *, int *)**

Recibe como primer argumento el arreglo de estructura a completar, antes de cargar el cuerpo de cada mail en el arreglo se debe verificar si el módulo es correcto, llamando a la función **identifica_modulo**, de ser correcto **incrementar en uno la cantidad de mails cargados**, cargar el cuerpo del mail y cargar el módulo. **PROGRAMAR**

➤ **char * identifica_modulo (char *)**

Se llama desde la función carga_ticket, recibe como argumento el cuerpo del mail. Devuelve el modulo si se reconoce como válido el mismo, “***” si el módulo no es correcto. **NO PROGRAMAR**

➤ **char * mayor_modulo (struct tickCol *, int , int *)**

Se llama desde el main, recibe como primer argumento el arreglo de estructura completo. El segundo argumento corresponde a la cantidad y el tercero a las ocurrencias del módulo. Determina que módulo tiene mayor ocurrencia, lo devuelve. En el caso de haber dos módulos de mayor ocurrencia, obtener el primer encontrado. **PROGRAMAR**

PROGRAMAR LA FUNCION main, carga_ticket y mayor_modulo.

EL PROGRAMA ES ITERATIVO.

ESCRIBIR DE UN SOLO LADO DE LA HOJA.