**Estacionamiento**

Suponga que posee un terreno en el Centro de Concepción, el cual, para poder sacar provecho, ha decidido convertirlo en un Lujoso estacionamiento.

Como usted es un prestigiado programador informático, ha decidido realizar un Software que le Ayude Con la administración de los Espacio y el cobro asociado a la estadía de cada vehículo.

Datos:

* El Cobro asociado a la estadía será por minuto, donde cada minuto de estancia cuesta $20.
* Los cupos máximos del estacionamiento Son 10.

El Sistema Deberá:

* Marcar La Entrada de un Automóvil, solicitando: Patente, Modelo y Hora de entrada (HH:MM).
* Marcar La salida de un automóvil, solicitando hora de salida(HH:MM) y entregando en pantalla el monto que el conductor debe pagar.
* Listar todos los autos que se encuentran estacionados.
* Listar cuantos son los cupos disponibles.
* Mostrar El Total de los cobros realizados.

Considere el uso de un Menú ***Switch***.

Validaciones:

* Deberá validar que el formato de Tiempo ingresado sea el correcto (HH:MM), es decir, el sistema debe mostrar que existe un error al ingresar un tiempo incorrecto y volverá a solicitarlo.
* En caso de que la hora de Ingreso y la Hora de salida sean las mismas o la hora de salida menor a la hora de ingreso, el sistema considerará como una estancia hasta el día siguiente.
  + Entrada 12:30 – Salida 12:30, el automóvil estuvo 24 horas (un día)
  + Entrada 12:30 – Salida 12:31, el automóvil estuvo 1 min.
  + Entrada 12:30 – Salida 12:10, el automóvil estuvo hasta las 12:10 del día siguiente (23 horas y 40 min).
* No pueden existir patentes duplicadas en el estacionamiento.

Utilice todas las Librerías que necesite en su software.

Si tiene a bien puede utilizar estas estructuras para realizar su Software o crear sus propias estructuras.

**typedef struct**

**{**

**char** patente**[6];**

**char** modelo**[20];**

tiempoinicio**;**

tiempofin**;**

**int** disponible**;**

**}**automovil**;**

**typedef struct**

**{**

**int** hora**;**

**int** minuto**;**

**}**tiempo**;**

Funciones útiles:

En caso de necesitar, borrar los datos de una Estructura, puede ayudarse de la siguiente función:

***void f\_limpiar\_struct(<Struct var>)***

***{***

***memset(&var, 0, sizeof(var));***

***}***

Esta función recibe como parámetro, una variable del tipo Struct.