Semestre 1 - Práctica 2
Sigue la reforma: el párquing

Índice

1. Introducción

Después de la reforma efectuada en los ascensores de los distintos edificios, en la Salle se quiere llevar a cabo también la reforma del párquing. Concretamente, se quiere implantar un nuevo sistema que permita controlar el acceso y salida de los vehículos, así como la ocupación total del párquing. Igual que en la reforma anterior, como garantía de seguridad y calidad esta tarea ha sido encargada a los alumnos de primer curso de la asignatura de Programación.

OBJETIVO

El objetivo final de esta práctica es acabar de afianzar el entorno de programación, aplicando los conceptos vistos a lo largo del semestre, tanto de algorítmica básica, como de estructuración y paso de parámetros y definición/uso de tipos propios. Concretamente, realizando esta práctica, se pretende :

- Consolidar las sentencias y algorítmica básica
- Consolidar los conocimientos acerca de la estructuración de un programa
- Consolidar el uso de procedimientos
- Consolidar el uso de funciones
- Consolidar el uso de parámetros
- Consolidar el uso de tipos de datos estructurados (arrays, cadenas y registros)
- Consolidar la definición y uso de tipos propios
- Consolidar la compilación manual de un proyecto en C

2. Funcionamiento

El programa empieza mostrando un mensaje de bienvenida y pidiendo que se introduzcan las tarifas actuales del párquing (ver Output 1).

```
Welcome to Parking LS!
Enter tariffs:
```

Output 1. El programa arranca dando la bienvenida y pidiendo las tarifas actuales del párquing.

La manera de definir las tarifas consiste en la lectura de una cadena de caracteres con el formato que se puede observar en el siguiente ejemplo:

`BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12`

Para cada tipo de vehículo aparecen las tarifas en las tres franjas disponibles: horario reducido, horario normal y hora punta (separados por el carácter '/'). Los horarios que se han definido para estas tres franjas son:

Horario reducido: de las 0:00h a las 7:59h, y de 21:00h a 23:59h

Hora punta: de las 10:00h a las 12:59h, y de 16:00h a 19:59h

Horario normal: resto de horas

El horario que se aplicará a cada vehículo dependerá exclusivamente de su hora de entrada al párking (ver apartado 2.2). Todas las tarifas serán valores enteros correspondientes a los céntimos de euro por minuto que cuesta la estancia en el párking.

Podemos suponer que el formato de la cadena es correcto, aunque el orden en el que aparecen los tipos de vehículos en la cadena puede variar. Así otra cadena perfectamente posible podría ser:

`TRUCKS:9/13/14#CARS:3/4/6#BIKES:1/2/3`

Una vez introducida la cadena, aparece el indicador de "prompt" con el texto "Parking LS>" (ver Output 2). A partir de este momento el programa queda a la espera de recibir comandos, con el formato que se explica en los siguiente apartados.

```
Welcome to Parking LS!
Enter tariffs: BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS>
```

Output 2. El programa lee las tarifas actuales del párquing y queda a la espera de recibir comandos.

2.1 Entrada de vehículos

Cuando un vehículo quiere entrar en el párquing se ejecuta el comando “enter” con el formato expresado en el siguiente ejemplo:

```
enter T 5645TYW 12:55
```

Después de la palabra “enter”, y separado por un espacio, se introduce un carácter que identifica el tipo de vehículo (B: bikes, C: cars, T: trucks). Si el tipo de vehículo no es correcto, se debe indicar que el comando es incorrecto (ver Output 3). A continuación, separado también por un espacio, se introduce la matrícula del coche (puede ser cualquier cadena de caracteres sin espacios de un máximo de 15 caracteres que siempre será correcta). Finalmente, se indica separado por un espacio la hora de entrada del vehículo.

Si el comando no está completo, es decir, falta alguno de los datos, se debe mostrar el mismo mensaje de error que antes (ver Output 4). Este mensaje de error también será el mismo si el usuario introduce un comando inventado o no definido en este documento.

```
Welcome to Parking LS!
Enter tariffs: BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS> enter K 6756FGT 12:55
(ERROR) Wrong command

Parking LS>
```

Output 3. El tipo de vehículo no es correcto y se muestra un mensaje de error.

```
Parking LS> enter
(ERROR) Wrong command

Parking LS> enter C
(ERROR) Wrong command

Parking LS> enter C 6756FGT
(ERROR) Wrong command

Parking LS>
```

Output 4. Diferentes ejemplos de errores por no introducir el comando completo.

También hay que comprobar que la hora introducida sea correcta. Esta hora debe encontrarse entre las 0:00 y las 23:59. Si esto no es así, se debe mostrar un mensaje de error concreto para este caso (ver Output 5).

```
Parking LS> enter C 6756FGT 12:64
(ERROR) Wrong time format

Parking LS> enter C 6756FGT 27:21
(ERROR) Wrong time format

Parking LS>
```

Output 5. Ejemplos donde el formato de la hora no es correcto y se muestra un error especial para este caso.

Finalmente, si se ejecuta el comando “enter” para un vehículo que ya se encuentra en el párquing se tiene que mostrar un error concreto para este caso (ver Output 6)

```
Welcome to Parking LS!
Enter tariffs: BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS> enter C 6766GTH 12:50

Parking LS> enter C 6766GTH 15:24
(ERROR) This vehicle is already in the parking!

Parking LS>
```

Output 6. Se introduce el comando “enter” sobre un vehículo que ya está en el párquing.

Únicamente se mostrará un error, es decir, en caso de que haya más de un error en el comando, se mostrará únicamente el primer error encontrado, según lo especificado en este enunciado.

Si el comando es correcto, entonces se almacena la información del vehículo en el sistema. Hay que tener en cuenta que el máximo de vehículos registrados en el párking viene definido por la siguiente constante:

```
#define MAX_VEHICLES 8
```

Si se superan el número de coches diferentes registrados en el párquing, se tiene que indicar esta circunstancia mediante el error indicado en el Output 7:

```
Welcome to Parking LS!
Enter tariffs:  BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS> enter C 6756FGT 12:55

Parking LS> enter B 1234TTT 13:34

Parking LS> enter T 1122BGT 13:46

Parking LS> enter T 8986GTT 17:00

Parking LS> enter T 1900FRW 19:25

Parking LS> enter C 1999FTW 20:20

Parking LS> enter B 1112TRW 21:25

Parking LS> enter C 6767BBX 21:39

Parking LS> enter C 9871CDF 22:23
(ERROR) No more vehicles are accepted

Parking LS>
```

Output 7. El número de vehículos registrados llega al máximo de los permitidos.

También hay que tener en cuenta que el programa sirve para la gestión de las entradas y salidas en un día concreto. Es decir, todas las horas corresponden al mismo día aunque un vehículo,

como veremos más adelante, puede entrar y salir más de una vez al día. El número máximo de veces que un vehículo puede entrar y salir, viene definido por la siguiente constante:

```
#define MAX_OPERATIONS 10
```

Aunque existe este límite, se puede suponer que un mismo vehículo no llegará nunca a este número de operaciones durante un día.

2.2 Salida de vehículos

Si un vehículo sale del párquing se ejecuta el comando “exit”, con el formato expresado mediante el siguiente ejemplo:

```
exit 5109GTY 13:14
```

Después del comando “exit” y separado por un espacio, se introduce la matrícula del coche que quiere salir. A continuación, después de otro espacio, se indica la hora de salida.

Del mismo modo que en la opción anterior, se tendrá que comprobar que el comando es correcto. Si falta algún dato o bien la hora no es correcta, se deberán mostrar los mismos errores que antes (ver Output 8).

```
Parking LS> exit
(ERROR) Wrong command

Parking LS> exit 6545TFG
(ERROR) Wrong command

Parking LS> exit 14:10
(ERROR) Wrong command

Parking LS>
```

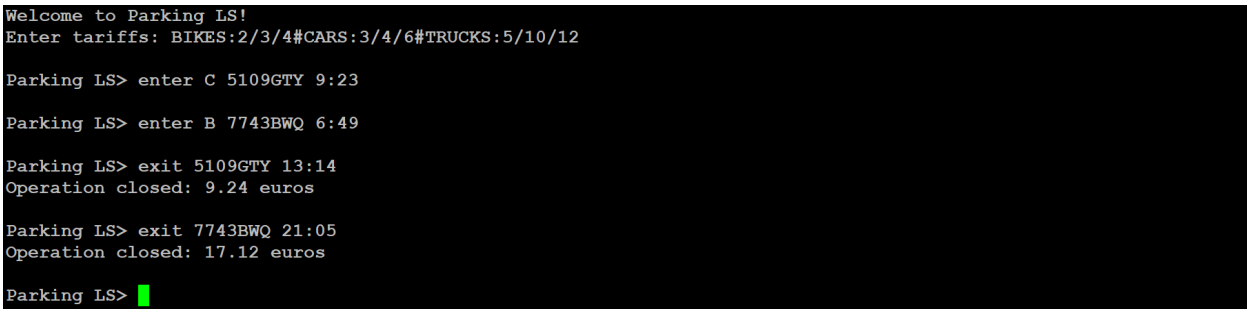
Output 8. Errores diferentes ocasionados por un comando “exit” incompleto.

Por otro lado, si el vehículo que se indica no se encuentra en el párquing (ya sea porque nunca entró o porque ya salió antes), se deberá mostrar un mensaje de error. También puede pasar que la hora de salida no sea posterior a la hora de entrada. En esta situación se deberá mostrar un error concreto para este caso. Ambos errores se pueden ver representados en el Output 9.

```
Welcome to Parking LS!
Enter tariffs: BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS> enter C 7744HGT 15:25

Parking LS> exit 8234BBC 17:45
(ERROR) This vehicle is not in the parking

Parking LS> exit 7744HGT 15:10
(ERROR) Incoherent exit time

Parking LS>
```

Output 9. Errores diferentes ocasionados por uso incorrecto del comando “exit”.

Finalmente, si el comando “exit” se utiliza correctamente, el programa cierra la operación y la guarda para posteriores consultas. En este caso por pantalla se muestra el importe de la estancia del vehículo, teniendo en cuenta el tipo de vehículo, los minutos transcurridos y la tarifa según la hora de entrada (ver Output 10). Todos los minutos cuestan lo mismo, según la tarifa definida por la hora de entrada.



Output 10. Dos vehículos entran y salen del párquing.

2.3 Consultas

El programa dispone del comando “show” para realizar consultas sobre los datos almacenados. Se pueden realizar dos operaciones distintas: consultar la ocupación del párquing y mostrar el detalles de las operaciones de un vehículo.

Para consultar la ocupación del párquing se ejecuta el siguiente comando:

```
show occupation
```

Con este comando se muestran todos los vehículos que se encuentran actualmente en el interior del párquing según el tipo de vehículo (el orden siempre será por orden del registro del vehículo). Si de alguna categoría no existen vehículos, se mostrarán los textos “No bikes”, “No cars” o “No trucks”, según el caso. Se puede ver un ejemplo completo, con el formato solicitado, en el Output 11.

```
Welcome to Parking LS!
Enter tariffs: BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS> enter B 6754BBV 11:15

Parking LS> enter C 8788CDF 13:27

Parking LS> enter C 9910FSD 14:25

Parking LS> enter C 9078FRS 17:19

Parking LS> exit 9910FSD 19:17
Operation closed: 11.68 euros

Parking LS> enter C 1909HJK 20:20

Parking LS> show occupation
Vehicles currently in the parking:
BIKES: 6754BBV
CARS: 8788CDF - 9078FRS - 1909HJK
TRUCKS: No trucks

Parking LS>
```

Output 11. Ejemplo de uso del comando “show occupation”.

Para consultar el detalle de un vehículo concreto, el comando a utilizar sigue el formato del siguiente ejemplo:

```
show detail 7645CDF
```

En este caso se tiene que comprobar que la matrícula del vehículo está registrada en el sistema. Si no es así, se tiene que mostrar un error al respecto (ver Output 12).

```
Welcome to Parking LS!
Enter tariffs: BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS> enter C 7636GTH 12:45

Parking LS> show detail 5634CDF
(ERROR) This vehicle never used the parking

Parking LS>
```

Output 12. Se intenta consultar el detalle de operaciones de un vehículo no registrado en el sistema.

En caso de que el vehículo se encuentre en el sistema, se mostrará el detalle de todas las operaciones de entrada/salida que ha realizado a lo largo del día (ver Output 13). Lo datos de las operaciones están separados por un tabulador (carácter '\t').

```
Welcome to Parking LS!
Enter tariffs: BIKES:2/3/4#CARS:3/4/6#TRUCKS:5/10/12

Parking LS> enter C 3454FFR 12:35

Parking LS> exit 3454FFR 13:47
Operation closed: 4.32 euros

Parking LS> enter C 3454FFR 16:20

Parking LS> exit 3454FFR 20:00
Operation closed: 13.20 euros

Parking LS> enter C 3454FFR 20:55

Parking LS> exit 3454FFR 23:23
Operation closed: 5.92 euros

Parking LS> show detail 3454FFR
Plate: 3454FFR
Type of vehicle: CAR
Operations:
        12:35   13:47   (4.32 euros)
        16:20   20:00   (13.20 euros)
        20:55   23:23   (5.92 euros)
Total paid: 23.44 euros

Parking LS>
```

Output 13. Se muestra el detalle de las operaciones de un vehículo concreto.

En caso de que el vehículo se encuentre dentro del párquing (todavía tiene pendiente de finalizar la operación actual), se indicará esta situación de la manera que se puede observar en el Output 14.

```
Parking LS> show detail 3454FFR
Plate: 3454FFR
Type of vehicle: CAR
Operations:
        12:35   13:47   (4.32 euros)
        16:20   20:00   (13.20 euros)
        20:55   23:23   (5.92 euros)
Total paid: 23.44 euros

Parking LS> enter C 3454FFR 23:30

Parking LS> show detail 3454FFR
Plate: 3454FFR
Type of vehicle: CAR
Operations:
        12:35   13:47   (4.32 euros)
        16:20   20:00   (13.20 euros)
        20:55   23:23   (5.92 euros)
        23:30   **:**   (**.** euros)
Total paid: 23.44 euros

Parking LS>
```

Output 14. El vehículo vuelve a entrar en el párquing y se muestra el detalle de sus operaciones.

Para el comando “show” también se deben mostrar los errores relacionados con su mal uso. En el Output 15, se pueden observar los casos de error.

```
Parking LS> show
(ERROR) Wrong command

Parking LS> show parking
(ERROR) Wrong command

Parking LS> show detail
(ERROR) Wrong command

Parking LS> show 6655FGT
(ERROR) Wrong command

Parking LS>
```

Output 15. Algunos de los errores causados por un mal uso del comando “show”.

2.4 Salir del programa

El comando “quit” finaliza la ejecución del programa, mostrando un mensaje de despedida al usuario (ver Output 16).

```
Parking LS> quit
See you later!
```

Output 16. El comando “quit” finaliza la ejecución del programa.

3. Consideraciones

Para la implementación de la práctica hay que tener en cuenta las siguientes consideraciones:

1. Se debe seguir el formato mostrado en los ejemplos.
2. En las salidas por pantalla se dejará como máximo una línea en blanco (observad los ejemplos a lo largo del enunciado).
3. La forma de gestionar errores deber ser la que se ha descrito en el enunciado.
4. El código debe estar correctamente comentado de modo que sea leíble sin dificultad. Comprobad que sigue la guía de estilos de la asignatura.
5. El código tiene que estar correctamente estructurado en procedimientos y/o funciones. Se pide un mínimo de 8 funciones o procedimientos (sin contar el main) durante el desarrollo de la práctica. Se valorará que el objetivo, diseño y uso de estas funciones/procedimientos sean correctos (significancia de las funciones o procedimientos, reaprovechamiento de código, paso de parámetros, etc).
6. Para que para la práctica sea aceptada es necesario que supere satisfactoriamente todos los tests de CodeRunner y tener una nota de Calidad del Software y de la memoria superior a 1.
7. El programa debería desarrollarse íntegramente en el entorno Linux de Matagalls, utilizando el editor vim para escribir el código C y el compilador gcc para la obtención del ejecutable correspondiente. Ahí la podéis probar antes de testearla en CodeRunner.
8. No se puede usar ninguna instrucción de C que no se haya explicado en clase.

4. Fechas de entrega y evaluación

La fecha de entrega de esta práctica para poder obtener la máxima calificación es el 14 de enero de 2024.

Para que la práctica se considere **entregada** se tendrán que cumplir las siguientes condiciones:

1. Ejecutar el código en el CodeRunner habilitado expresamente y superar satisfactoriamente los tests.
2. Entregar en el pozo correspondiente un archivo .c con el código de la práctica.
3. Entregar una memoria final de la práctica, incluyendo:
    a. Portada e índice
    b. Breve resumen del enunciado de la práctica
    c. Explicación de cómo se ha estructurado la práctica (procs/funcs)
    d. Principales dificultades y soluciones aplicadas
    e. Conclusiones finales
    f. Estimación del tiempo dedicado para realizar la práctica

Se recuerda que la práctica será evaluada de la siguiente manera:

- Ejecución: Tiene un peso del 60% y evalúa el correcto funcionamiento de la práctica. Será la calificación obtenida en los tests de CodeRunner.

- Calidad del SW: Tiene un peso del 20% y se evaluará la calidad del código entregado y el seguimiento de la Guía de Estilo de Programación que tenéis disponible en el estudy.

- Memoria: Tiene un peso del 20% y se evaluará revisando el contenido de todos los puntos indicados.

La evaluación de la práctica será sobre 10 puntos, 8 puntos o 6 puntos, en función de cuando se entregue la práctica (y sea aceptada):

Sobre 10 puntos:	hasta el 14 de enero de 2024, a las 23:59h.
Sobre 8 puntos:	desde el 15 de enero al 9 de junio de 2024, a las 23:59h.
Sobre 6 puntos:	desde el 10 de junio de 2024 al 7 de julio de 2024, a las 23:59h.