

Semestre 1 - Práctica 1

La reforma



Índice

1. Introducción.....	3
2. Funcionamiento.....	4
2.1 Opción 1: Use Elevator	5
2.2 Opción 2: Statistics	7
2.3 Opción 3: Elevators inspection.....	8
2.4 Opción 4: Quit	12
3. Consideraciones.....	13
4. Fechas de entrega y evaluación	14

1. Introducción

En la Salle se está estudiando llevar a cabo una serie de reformas la primera de las cuales afecta a los ascensores de los diferentes edificios. Concretamente, se quiere implantar un sistema común para todos ellos que consiste en la puesta en marcha de 3 nuevos ascensores por edificio, denominados Alpha, Beta y Gamma. Como este sistema es novedoso, se quiere hacer un primer diseño y verificación de su software de control. Como garantía de seguridad esta tarea ha sido encargada a los alumnos de primer curso de la asignatura de Programación.

OBJETIVO

El objetivo final de esta práctica es poder entrar en contacto con el entorno de programación y poner en práctica los conceptos de algorítmica básica ya vistos en clase. Concretamente, se pretende :

- Consolidar la lectura de entrada por teclado
- Consolidar la escritura de datos por pantalla
- Consolidar el uso de sentencias condicionales
- Consolidar el uso de sentencias alternativas
- Consolidar el uso de sentencias iterativas
- Consolidar la compilación manual de un proyecto en C

2. Funcionamiento

El programa empieza mostrando un mensaje de bienvenida y a continuación muestra las diferentes opciones disponibles mediante un menú. Estas opciones son: “Use Elevator”, “Statistics”, “Elevators inspection” y “Quit” (ver Output 1).

```
Welcome to our building!  
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit  
Enter option: █
```

Output 1. El programa arranca dando la bienvenida y mostrando las opciones disponibles.

Después de ejecutar cualquier opción del menú (excepto la opción “Quit”), el programa debe volver al menú principal. Como veremos más adelante, si se ejecuta la opción “Quit”, el programa debe finalizar.

El usuario puede introducir cualquiera de las opciones mediante una letra mayúscula o minúscula (case insensitive).

Si el usuario introduce una opción incorrecta, el programa debe mostrar un error indicando esta circunstancia (Ver Output 2). A continuación se vuelve a mostrar el menú con las mismas opciones que antes.

```
Welcome to our building!  
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit  
Enter option: P  
(ERROR) Invalid option  
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit  
Enter option: █
```

Output 2. El programa muestra un error en caso que la opción no sea válida.

2.1 Opción 1: Use Elevator

La opción “Use Elevator” se ejecuta cuando un usuario quiere utilizar uno de los ascensores de un edificio. Como se ha comentado en la introducción, cada edificio dispone de 3 ascensores denominados Alpha, Beta y Gamma. Hay que tener en cuenta también que los edificios donde estarán situados los ascensores siempre constarán de ocho (8) plantas, etiquetadas de la siguiente manera:

Ático (Attic)
Planta 5
Planta 4
Planta 3
Planta 2
Planta 1
Planta baja (Ground)
Sótano (Basement)

Cada vez que se quiere usar el ascensor, en el panel de control del ascensor se introduce la planta donde se encuentra el usuario y su planta destino. El usuario lo indica mediante un carácter, que puede ser una letra mayúscula para el Ático (A), Planta Baja (G) o Sótano (B), o bien un dígito para el resto de plantas. Se pedirá al usuario cada uno de los datos hasta que ambos sean completamente correctos (ver Output 3).

```
Welcome to our building!
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? 9
From? 5
To? J
To? 0
To? █
```

Output 3. Se pide la planta origen y destino al usuario. Si algún dato no es correcto, se vuelve a pedir.

Para evitar que los ascensores se desplacen innecesariamente y para optimizar el tiempo de espera del usuario, el ascensor que se activará será aquel que se encuentre más cerca de la planta origen. En caso de que dos o más ascensores se encuentren a igual distancia de la planta origen del usuario (positiva o negativa), se activará aquel que haya recorrido menos plantas con carga hasta el momento. Es decir, para el cálculo de las plantas recorridas no se contabilizarán aquellas que tenga que hacer el ascensor para ir a buscar a los usuarios sino únicamente las realizadas en los trayectos marcados por los usuarios. Si aún así, se mantuviera el empate, el ascensor Alpha se activaría antes que el Beta, y éste antes que el Gamma. Inicialmente, cuando arranque el programa, todos los ascensores se encontrarán en la Planta Baja (G).

Si el usuario indica como planta destino la misma que la del origen, el sistema detecta esta circunstancia y muestra un error por pantalla (ver Output 4).

```
Welcome to our building!

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: a

From? 5
To? 5
(ERROR) Are you joking?

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: █
```

Output 4. El usuario indica el mismo destino que origen y el sistema muestra un error por pantalla.

Una vez que los datos son correctos, el sistema escoge el ascensor asignado al usuario y el nombre de este se muestra por pantalla. A continuación se muestra el recorrido que hace el ascensor para ir a buscar al usuario (únicamente si necesita moverse). Una vez allí, se indica si el ascensor va a subir o bajar y, en la misma línea, se muestra el recorrido realizado por el ascensor, indicando cada una de las plantas por las que pasa y separando cada carácter por dos puntos seguidos (..). Se puede observar su ejecución en el Output 5.

```
Welcome to our building!

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: a

From? 4
To? A
Elevator Alpha assigned
G..1..2..3..4 Going up! 4..5..A

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? 2
To? B
Elevator Beta assigned
G..1..2 Going down! 2..1..G..B

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: a

From? 5
To? 2
Elevator Alpha assigned
A..5 Going down! 5..4..3..2

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: █
```

Output 5. Ejemplos de uso de los ascensores.

2.2 Opción 2: Statistics

La opción “Statistics” permite obtener un conjunto de estadísticas para cada uno de los ascensores existentes. Concretamente, para cada ascensor, se muestra el número de viajes realizados, indicando entre paréntesis cuántos han sido de subida y cuántos de bajada. También, en la misma línea se muestra el número de plantas recorridas en total (Output 6).

```
Welcome to our building!

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: a

From? 4
To? A
Elevator Alpha assigned
G..1..2..3..4 Going up! 4..5..A

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? 1
To? 6
To? 5
Elevator Beta assigned
G..1 Going up! 1..2..3..4..5

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? G
To? B
Elevator Gamma assigned
Going down! G..B

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? 4
To? G
Elevator Beta assigned
5..4 Going down! 4..3..2..1..G

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: B

Elevators statistics:

ALPHA: 1 rides (1-0) and 2 floors traveled
BETA: 2 rides (1-1) and 8 floors traveled
GAMMA: 1 rides (0-1) and 1 floors traveled

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: █
```

Output 6. Se utilizan varios ascensores y a continuación se ejecuta la opción “Statistics”.

2.3 Opción 3: Elevators inspection

La opción 3 del programa permite detectar cuáles son los ascensores que necesitan ser inspeccionados por haber recorrido ya muchas plantas. El número de plantas que indica que el ascensor necesita una revisión viene definido por la siguiente constante que se deberá declarar en el programa:

```
#define FLOORS_FOR_INSPECTION 12
```

Por otro lado, el programa también irá calculando el precio de las inspecciones realizadas hasta el momento. El precio base de una inspección viene marcado por la siguiente constante, que también se deberá definir:

```
#define INSPECTION_PRICE 150
```

Para calcular el precio total de la inspección de un ascensor, se deberá multiplicar el precio base por la cantidad de veces que se ha superado el nivel de inspección. Es decir, si por ejemplo un ascensor ha recorrido 44 plantas, el precio de la inspección será de 3 veces el precio base de la inspección (150×3) ya que ha superado más de 3 veces el número de plantas recorridas para activar la inspección. Es decir, cuantas más plantas haya recorrido sin realizar la inspección, más difícil y más cara será.

Cuando se active la opción “Elevators inspection”, se mostrarán los ascensores que requieren de inspección por haber superado el número mínimo de plantas recorridas. A continuación, se preguntará al usuario si efectivamente quiere efectuar la inspección de los ascensores. Si el usuario introduce una letra ‘Y’ (mayúscula o minúscula) entonces se procederá a la inspección y se mostrará para cada ascensor un mensaje, indicando el total de euros gastados en esa inspección. Finalmente se mostrará el total de gasto histórico (de todas la revisiones ya hechas en el pasado). Una vez hecha la inspección, se resetea el número de plantas recorridas de los ascensores que han pasado la inspección, poniéndolo a 0 y borrando los valores actuales. Se puede observar un ejemplo completo en el Output 7.

En caso que el usuario acabe indicando que no quiere realizar la inspección de los ascensores que la necesitan (opción ‘N’ mayúscula o minúscula), entonces se mostrará únicamente el gasto acumulado histórico y no se realizará ninguna acción adicional (ver Output 8).

Si el usuario no introduce una letra ‘Y’ o ‘N’ (mayúscula o minúscula) se volverá a pedir este dato (ver Output 8).


```
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: B

Elevators statistics:

ALPHA: 3 rides (1-2) and 15 floors traveled
BETA: 4 rides (2-2) and 27 floors traveled
GAMMA: 2 rides (1-1) and 10 floors traveled

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: C

Elevator Alpha: inspection required
Elevator Beta: inspection required

Perform inspection? y
Elevator Alpha inspected! (150 euros)
Elevator Beta inspected! (300 euros)

Accumulated expense: 450 euros

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: B

Elevators statistics:

ALPHA: 3 rides (1-2) and 0 floors traveled
BETA: 4 rides (2-2) and 0 floors traveled
GAMMA: 2 rides (1-1) and 10 floors traveled

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: █
```

Output 7. Se muestra primero las estadísticas actuales de los ascensores. Después se pasa la inspección de dos ascensores y se vuelven a mostrar las estadística (se puede observar que el número de plantas recorridas de los ascensores inspeccionados ha sido reseteado).

```
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: C

Elevator Alpha: inspection required

Perform inspection? p
Perform inspection? N

Accumulated expense: 300 euros

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: █
```

Output 8. En este otro caso el ascensor Alpha necesita de inspección, pero el usuario descarta realizarla.

Si ningún ascensor necesita revisión, entonces se mostrará el mensaje correspondiente, indicando también el total de gasto de inspecciones (Output 9).

```
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: C

No required inspection

Accumulated expense: 0 euros

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: █
```

Output 9. Ningún ascensor necesita revisión. La captura que se puede ver es la resultante de ejecutar esta opción al arrancar el programa.

En esta página y la siguiente se muestra un ejemplo de ejecución completo (Ouput 10):

```
Welcome to our building!

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? b
From? B
To? A
Elevator Alpha assigned
G..B Going up! B..G..1..2..3..4..5..A

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: a

From? 5
To? B
Elevator Alpha assigned
A..5 Going down! 5..4..3..2..1..G..B

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? 1
To? 4
Elevator Beta assigned
G..1 Going up! 1..2..3..4

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? G
To? 1
Elevator Gamma assigned
Going up! G..1

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? B
To? 6
To? 5
Elevator Alpha assigned
Going up! B..G..1..2..3..4..5

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? 5
To? B
Elevator Alpha assigned
Going down! 5..4..3..2..1..G..B

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? 4
To? B
Elevator Beta assigned
Going down! 4..3..2..1..G..B

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? B
To? A
Elevator Beta assigned
Going up! B..G..1..2..3..4..5..A
```

```
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: B

Elevators statistics:

ALPHA: 4 rides (2-2) and 25 floors traveled
BETA: 3 rides (2-1) and 15 floors traveled
GAMMA: 1 rides (1-0) and 1 floors traveled

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: C

Elevator Alpha: inspection required
Elevator Beta: inspection required

Perform inspection? y
Elevator Alpha inspected! (300 euros)
Elevator Beta inspected! (150 euros)

Accumulated expense: 450 euros

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? A
To? G
Elevator Beta assigned
Going down! A..5..4..3..2..1..G

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: A

From? G
To? A
Elevator Beta assigned
Going up! G..1..2..3..4..5..A

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: B

Elevators statistics:

ALPHA: 4 rides (2-2) and 0 floors traveled
BETA: 5 rides (3-2) and 12 floors traveled
GAMMA: 1 rides (1-0) and 1 floors traveled

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: C

Elevator Beta: inspection required

Perform inspection? Y
Elevator Beta inspected! (150 euros)

Accumulated expense: 600 euros

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: B

Elevators statistics:

ALPHA: 4 rides (2-2) and 0 floors traveled
BETA: 5 rides (3-2) and 0 floors traveled
GAMMA: 1 rides (1-0) and 1 floors traveled

A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit
Enter option: Q

See you later!
```

Output 10. Ejemplo de ejecución completo, usando todas las opciones del programa.

2.4 Opción 4: Quit

La opción 4 del programa muestra un mensaje de despedida al usuario y finaliza la ejecución del programa (ver Output 11).

```
A. Use Elevator | B. Statistics | C. Elevators inspection | Q. Quit  
Enter option: Q  
See you later!
```

Output 11. La opción "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 los errores es estrictamente la que pide el enunciado. Si se dice que debe volver a introducirse el dato no se aceptará que por ejemplo vuelva al menú inicial.
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. 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.
6. 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.
7. 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 26 de noviembre de 2023.

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. Diagrama de actividades de la práctica
 - 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 26 de noviembre de 2023, a las 23:59h. |
| Sobre 8 puntos: | desde el 27 de noviembre 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. |