

**FORMATO ENTREGA**

Para cada ejercicio se debe generar su correspondiente código.py y un documento con capturas de pantallas donde se compruebe la correcta ejecución del mismo (pruebas realizadas para su validación).

Nombre de los ficheros: iniciales\_ejx.extensión

Ejemplo. Marta Rodríguez Sánchez → **MRS\_ej1.py // MRS\_ej1.doc**

En el ejercicio 3 se marca la opción elegida → **MRS\_ej3A.py // MRS\_ej3A.doc**

Agrupar todos los archivos creados en una carpeta, comprimirlos y subir al aula virtual.

Nombre carpeta: iniciales\_ex\_di\_dam → **MRS\_ex\_di\_dam**

**IMPORTANTE.** - Para poder realizar la entrega, se debe notificar al docente que se ha concluido para su supervisión.

**DURACIÓN: 17:40 – 20:00**

### Ejercicio 1 (2,5 puntos)

Pide una cadena y dos caracteres por teclado (valida que sea un carácter), sustituye la aparición del primer carácter en la cadena por el segundo carácter.

### Ejercicio 2 (3,5 puntos)

Escribe un programa que pida el límite inferior y superior de un intervalo. Si el límite inferior es mayor que el superior lo tiene que volver a pedir.

A continuación se van introduciendo números hasta que introduzcamos el 0. Cuando termine el programa dará las siguientes informaciones:

- La suma de los números que están dentro del intervalo (intervalo abierto).
- Cuantos números están fuera del intervalo.
- Informa si hemos introducido algún número igual a los límites del intervalo.

### Ejercicio 3 (4 puntos)

Elegir una de las dos siguientes opciones:

**A.** Crear un programa con el siguiente menú:

1. Añadir número: pide un número y lo añade al final de la lista.
2. Añadir número a la lista en una posición: pide un número y una posición, y si la posición existe en la lista lo añade a ella (la posición empieza en 1).
3. Longitud de la lista: muestra el número de elementos de la lista.
4. Eliminar el último número: muestra el último número de la lista y lo borra.
5. Eliminar un número: pide una posición, y si la posición existe en la lista lo borra de ella (la posición se pide a partir de 1).
6. Contar apariciones: pide un número y dice veces aparece en la lista.
7. Posiciones de un número: pide un número y dice en que posiciones está (contando desde 1).
8. Mostrar números: muestra los números de la lista.
9. Salir.

**B.** Escribir un programa que implemente una agenda. En la agenda se podrán guardar nombres y números de teléfono. El programa nos dará el siguiente menú:

1. Añadir/modificar: pide un nombre. Si el nombre se encuentra en la agenda, debe mostrar el teléfono y, opcionalmente, permitir modificarlo si no es correcto. Si el nombre no se encuentra, debe permitir ingresar el teléfono correspondiente.
2. Buscar: pide una cadena de caracteres, y muestra todos los contactos cuyos nombres comiencen por dicha cadena.
3. Borrar: pide un nombre y si existe, pregunta si se quiere borrar de la agenda.
4. Listar: muestra todos los contactos de la agenda.

Implementar el programa con un diccionario.