

- 10.11. Escribe un programa que solicite al usuario el nombre de un fichero de texto y muestre su contenido en pantalla. Si no se proporciona ningún nombre de fichero, la aplicación utilizará por defecto prueba.txt.
- 10.12. Diseña una aplicación que pida al usuario su nombre y edad. Estos datos deben guardarse en el fichero datos.txt. Si este fichero existe, deben añadirse al final en una nueva línea, y en caso de no existir, debe crearse.
- 10.13. Implementa un programa que lea dos listas de números enteros no ordenados de sendos archivos con un número por línea, los reúna en una lista única y los guarde en orden creciente en un tercer archivo, de nuevo uno por línea.
- 10.14. Escribe un programa que lea un fichero de texto llamado carta.txt. Tenemos que contar los caracteres, las líneas y las palabras. Para simplificar supondremos que cada palabra está separada de otra por un único espacio en blanco o por un cambio de línea.
- 10.15. En el archivo numeros.txt disponemos de una serie de números (uno por cada línea). Diseña un programa que procese el fichero y nos muestre el menor y el mayor.
- 10.16. Un libro de firmas es útil para recoger los nombres de todas las personas que han pasado por un determinado lugar. Crea una aplicación que permita mostrar el libro de firmas e insertar un nuevo nombre (comprobando que no se encuentre repetido). Llamaremos al fichero firmas.txt.
- 10.17. En Linux disponemos del comando more, al que se le pasa un fichero y lo muestra poco a poco: cada 24 líneas. Implementa un programa que funcione de forma similar.
- 10.18. Escribe la función Integer [] leerEnteros (String texto) , al que se le pasa una cadena y devuelve una tabla con todos los enteros que aparecen en ella.
- 10.19. Un encriptador es una aplicación que transforma un texto haciéndolo ilegible para aquellos que desconocen el código. Diseña un programa que lea un fichero de texto, lo codifique y cree un nuevo archivo con el mensaje cifrado. El alfabeto de codificación se encontrará en el fichero codec.txt. Un ejemplo de codificación de alfabeto es:
Alfabeto: abcdefghijklmnopqrstuvwxyz
Cifrado: emsrcyjnfxiwtakozdlqvbhupg
- 10.20. Algunos sistemas operativos disponen de la orden comp, que compara dos archivos y nos dice si son iguales o distintos. Diseña esta orden de forma que, además, nos diga en qué línea y carácter se encuentra la primera diferencia. Utiliza los ficheros texto1.txt y texto2.txt.

10.21. Diseña una pequeña agenda, que muestre el siguiente menú:

1. Nuevo contacto.
2. Buscar por nombre.
3. Mostrar todos.
4. Salir.

En ella, guardaremos el nombre y el teléfono de un máximo de 20 personas.

La opción 1 nos permitirá introducir un nuevo contacto siempre y cuando la agenda no esté llena, comprobando que el nombre no se encuentra insertado ya. La opción 2 muestra todos los teléfonos que coinciden con la cadena que se busca. Por ejemplo, si tecleamos «Pe», mostrará el teléfono de Pedro, de Pepe y de Petunia. La opción 3 mostrará un listado con toda la información (nombres y teléfonos) ordenados alfabéticamente por el nombre. Por último, la opción 4 guarda todos los datos de la agenda en el archivo agenda.txt. La próxima vez que se ejecute la aplicación, si hay datos guardados, se cargarán en memoria.

10.22. Crea con un editor de texto el fichero deportistas.txt, donde se recogen los datos de un grupo de deportistas, uno en cada línea. Aparecerá el nombre completo, seguido de la edad, el peso y la estatura. La primera línea será el encabezamiento con los nombres de los campos. El documento tendrá la siguiente forma:

10.23. Nombre Edad Peso Estatura Juan Pedro Pérez Gómez 25 70,5 1,80 Ana Ruiz del Val 23 60 1,75

10.24. Implementa un programa donde se cree un flujo de texto de entrada, a partir del cual, usando un objeto Scanner, se leerán los datos de los deportistas, que se mostrarán por pantalla. Al final aparecerán los valores medios de la edad, el peso y la estatura.