

EJERCICIOS PROPUESTOS PARA LOS MÓDULOS DE PROGRAMACIÓN

CICLOS FORMATIVOS DE INFORMÁTICA

CENTRO DE PROFESORES Y RECURSOS
CÁCERES

**Ejercicios Propuestos para los Módulos de Programación
Ciclos Formativos de Informática**

Inés M^a García Cabañas, Fernando Sánchez Rubio, Enrique Guzmán Romero

Edita: Centro de Profesores y Recursos de la Provincia de Cáceres

Imprime: **Imprenta Luengo.**

C/. Río Miño, s/n

Telf. y Fax: 927 53 29 28

10300 Navalmoral de la Mata

(Cáceres)

I.S.B.N.: 84-688-2893-9

Depósito Legal: CC-252-2003

PRESENTACIÓN

El libro que ahora tiene en sus manos forma parte de una serie de publicaciones que poseen un denominador común: han sido concebidas como expresión y muestra de las experiencias realizadas por profesores de centros educativos de nuestra Comunidad Autónoma.

Son fruto del trabajo desarrollado por estos docentes en el seno de grupos de trabajo, seminarios, proyectos de formación en centros y otras actividades de formación en las que participan, dependiente de los Centros de Profesores y Recursos (CPR,s) de la provincia de Cáceres.

Esta serie de publicaciones intenta dar a conocer a la comunidad educativa en general, y a los profesores en particular, el contenido de algunas experiencias significativas como aportación cercana del profesorado hacia sus propios compañeros.

Con ellas se pretende difundir las innovaciones, investigaciones y experiencias que se desarrollan en los centros educativos, fomentadas y asesoradas desde los Centros de Profesores y Recursos.

Sirvan estos materiales como reconocimiento a todos los maestros y profesores que han contribuido con su esfuerzo, trabajo y dedicación a que la apuesta por la divulgación de materiales educativos surgidos de la práctica diaria se convierta en una realidad.

Septiembre de 2003

Los Centros de Profesores y Recursos de la provincia de Cáceres.

PRÓLOGO

A través de esta serie de ejercicios se pretende trabajar con distintas estructuras y tipos de datos. No están dirigidos hacia un lenguaje concreto, por lo que cada uno puede enfocarlo bien hacia PASCAL, C, C++, etc.

Principalmente está orientado hacia los C.F.G.S. de la familia de Informática en los módulos en los que se trabaje la programación. Son, siempre, una guía orientativa, de forma que los profesores tengan un punto inicial, a partir del cual puedan confeccionar los ejercicios que estimen más convenientes.

Se trabajarán:

Estructuras básicas	Ejercicios 1 a 23 (Pág. 7)
Vectores y matrices	Ejercicios 24 a 35 (Pág. 12)
Ordenación de vectores y matrices	Ejercicios 36 a 38 (Pág. 16)
Tratamiento de cadenas	Ejercicios 39 a 48 (Pág. 19)
Estructuras o Registros	Ejercicios 49 a 50 (Pág. 21)
Ficheros de texto	Ejercicios 51 a 59 (Pág. 23)
Ficheros con registros	Ejercicios 60 a 75 (Pág. 25)
Punteros y Estructuras dinámicas	Ejercicios 76 a 89 (Pág. 45)
Programación orientada a objeto	Ejercicios 90 a 98 (Pág. 49)

Es nuestro deseo que os sea de utilidad.

Inés M^a Gracia Cabañas
Fernando Sánchez Rubio
Enrique Guzmán Romero

ESTRUCTURAS BÁSICAS

- 1.- Programa que lea 3 números y los escriba ordenados de forma ascendente.

- 2.- Programa que recibe como dato de entrada una hora expresada horas, minutos y segundos y escribe la nueva hora transcurrido 1 segundo. Controlar el valor de las entradas.

- 3.- Programa que lee un n° de notas con valores de 0 a 10, que termina con el valor -1 , y nos dice si hubo o no alguna con el valor 10.

- 4.- Programa que calcula el factorial de un n° entero y positivo.

- 5.- Programa que lee un n° entero X y otro n° entero positivo N y calcula la $n^{\text{ésima}}$ potencia de X .

- 6.- Programa que calcula el producto de dos números enteros y positivos mediante sumas sucesivas.

- 7.- Programa que obtiene el cociente y el resto de dos números enteros y positivos mediante restas sucesivas. A y B son las variables para leer datos de entrada (dividendo y divisor), C es la variable para calcular el cociente. En A se guardará el resto. (Nota: La división entre 0 no está definida).

8.- Programa que dado dos intervalos cerrados $[a, b]$ y $[c, d]$ de la recta real devuelva la intersección de ambos intervalos.

9.- Algoritmo que diga si un número entero positivo es primo o no (se va dividiendo por 2, 3, 5, 7..... hasta $N-1$). Si ninguna de las divisiones es exacta y llegamos a una en la que el cociente es menor o igual que el divisor, el n° es primo.

10.- Dado un número entero positivo N , escribe un programa C que determine cuántos dígitos se necesitan para escribir el valor N . Por ejemplo, 99 necesita 2 dígitos, 115 necesita 3, para 1000 la salida es 4.

11.- Un número natural se denomina perfecto cuando es igual a la suma de todos sus divisores, primos o no excepto él mismo. Por ejemplo $128 = 1 + 2 + 4 + 7 + 14$. Escribe un programa C que averigüe si un número natural positivo es perfecto o no.

12.- Escribe un programa C que lea como dato de entrada una fecha expresada en días (1-31), mes (1-12) y año (en n°) y nos dice la fecha que será al día siguiente. Se supone que febrero siempre tiene 28 días.

13.- Programa que genera la lista de los N primeros números primos, N es el dato de entrada.

14.- Programa que evalúa un polinomio de grado N . Los datos de entrada son el valor de la variable, el grado y los valores de los coeficientes.

15.- Programa que calcule el m.c.d. de dos números A y B ($A \geq B$). Algoritmo:

1. Hacer $dd = A$ y $d = B$.
2. Calcular q cociente entero de la división de dd por d y el resto r .
3. Si $r \neq 0$ hacer $dd = d$ y $d = r$ e ir al paso 1°. Si $r = 0$, d es el m.c.d.

16.- Programa que lea dos números $N1$ y $N2$ enteros positivos y obtiene su mínimo común múltiplo. (Se sabe que el mínimo común múltiplo de dos números es igual a su producto $N1 * N2$ dividido entre su m.c.d).

17.- Escribe un programa que lea un número e imprime un triángulo de números de la forma siguiente. Si el número leído es 4, imprimirá:

```
1
22
333
4444
```

18.- En el supermercado de la esquina colocan las latas de conservas apiladas triangularmente. Esto trae bastantes problemas a la hora de hacer los pedidos, ya que quieren un número de latas de manera que no sobre ni falte ninguna. Así, 6 latas se pueden apilar triangularmente de la siguiente manera.

```
  *
 * *
* * *
```

Si tuvieran 7 latas, no podrían apilarse de forma triangular, ya que sobraría una. Escribir un programa C, que dado un número natural, compruebe si es adecuado para montar pilas triangulares.

19.- En el mismo supermercado quieren montar también pilas del mismo número de latas de alto que de ancho.

```
* * *  
* * *  
* * *
```

Ahora bien no todas las cantidades de latas que pueden apilarse triangularmente pueden ponerse en estructuras cuadradas. Diseñas un programa C que indique si un número natural es válido para realizar los dos tipos de estructuras. (Un ejemplo válido es el número 36).

20.- Dado un número natural N, se calcula la raíz digital de N sumando los dígitos que lo componen. El proceso se repite sobre el nuevo número hasta que el resultado obtenido tiene un sólo dígito. Este último número es la raíz digital del número N.

Ejemplo: $347 \rightarrow 3 + 4 + 7 = 14 \rightarrow 1 + 4 \rightarrow 5 \rightarrow$ Raíz digital (347) = 5.

Escribir un programa C que calcule la raíz digital de un número.

21.-Escribe un programa C que, dado un número entero positivo y lo devuelva al revés. Por ejemplo, si el número de entrada es 1234, la rutina debería devolver 4321.

22.- Escribir un programa que lea un número natural de 4 cifras, no todos iguales. Implementa la función Mayor(N), que calcula el mayor número que se puede formar con las cifras de N, y la función Menor(N) que calcula el menor número que se puede formar con las cifras de N.

23.- Construir un algoritmo que permita multiplicar dos números enteros positivos empleando el método denominado MULTIPLICACIÓN RUSA. Este método permite calcular el producto de $M \cdot N$ de la siguiente manera:

En pasos sucesivos se divide M por 2 (división entera) y se multiplica N por 2. Este proceso se repite hasta que M es 0. El resultado de la multiplicación deseada se obtiene acumulando aquellos valores sucesivos de N para los cuales el valor de M es impar:

Ejemplo 1: $31 \cdot 27$

N	M	Acumulado
31*	27	31
62*	13	31+62
124	6	31+62
248*	3	31+62+248
496*	1	31+62+248+496
992	0	

Ejemplo 2: $25 \cdot 6$

N	M	Acumulado
25	6	0
50*	3	0+50
100*	1	0+50+100
200	0	

VECTORES Y MATRICES

24.- Programa que lee 50 números enteros sobre un vector y obtiene e imprime cuáles son el mayor y el menor número almacenados y cuántas veces se repiten ambos.

25.- Con el mismo vector del ejercicio anterior. Programa que calcula e imprime las sumas de las componentes de índice par y las de índice impar.

26.- Con el mismo vector del ejercicio anterior. Programa que invierte dicho vector imprimiéndolo.

27.- Dado un número entero positivo de 10 cifras. Programa que compruebe si un número es capicúa utilizando un vector de 10 componentes.

28.- Programa que carga una matriz de 5 filas y 10 columnas con números enteros imprimiendo los valores máximo y mínimo y sus posiciones dentro de la tabla.

29.- Con la misma matriz del ejercicio anterior. Programa que obtiene e imprime su matriz traspuesta.

30.- Programa que genera e imprime una matriz unitaria de orden N. Una matriz unitaria de orden N es la que tiene N filas y N columnas con todas sus componentes a 0, excepto las de su diagonal principal que están a 1.

31.- Programa que lee un vector de N elementos y rota todas sus componentes un lugar hacia su derecha. Teniendo en cuenta que la última componente se ha de desplazar al primer lugar.

32.- Programa que imprime un cuadrado latino de orden N. Un cuadrado latino de orden N es una matriz cuadrada en la que la primera fila contiene los N primeros números naturales y cada una de las siguientes N-1 filas contiene la rotación de la fila anterior un lugar a la derecha.

Ejemplo: Cuadrado latino de orden 4.

```
1 2 3 4
4 1 2 3
3 4 1 2
2 3 4 1
```

33.- Generar un número de cuatro cifras (no debe repetirse ninguna de ellas).

Realizar un programa para adivinar el número anterior. Se introducirá otro número de cuatro cifras (sin repetir ninguna) y se comparará con el generado anteriormente. Si coincide la cifra y la posición se indicará con el signo "+", si coincide sólo la cifra (y no la posición) se indicará con el signo "-".

Ej:

7 2 1 4 (número generado, que no aparecerá en pantalla)

2 0 6 4 (número que introduce el usuario)

A la derecha del número introducido aparecerá un signo "+" (el 4 está acertado y además está en su sitio) y un signo "-" el 2 está acertado, pero no está en su sitio.

Se darán diez oportunidades para adivinar el número, si no se consigue en este plazo se mostrará dicho por pantalla.

Tanto si se adivina el número como si no se dará la posibilidad de jugar de nuevo.

34.- Generar 30 números entre el 1 y el 99 y dar como resultados de salida:

- a) Número más alto.
- b) Número más bajo.
- c) Media aritmética.
- d) Moda.
- e) Mediana.
- f) Cantidad de 1,2...10 y sus porcentajes.

Cantidad de 1,2,3...10	Porcentaje

35.- Obtener diez números (0..9) mediante el generador de números aleatorios. Con estos diez números se formarán dos de cinco cifras cada uno.

Sumar los dos números de cinco cifras obtenidos. La suma se realizará cifra a cifra (unidades, decenas, centenas, etc.), apareciendo los sumandos y el resultado en letra, también se indicará en letra si se llevan una y aparecerá un "1" encima del sumando correspondiente.

Ej:

$$\begin{array}{r}
 1 \\
 7\ 9\ 1\ 4\ 6 \quad \text{SEIS Y} \\
 + 1\ 6\ 3\ 7\ 5 \quad \text{CINCO} \\
 \hline
 1 \quad \text{ONCE} \quad \text{y me llevo una}
 \end{array}$$

Pulse <Intro> para continuar

Al pulsar Intro pasarán a sumarse las dos cifras siguientes.

ORDENACIÓN DE VECTORES Y MATRICES

36.- Llenar una tabla de 10 posiciones con números enteros comprendidos entre el 1 y el 99. Ordenar dicha tabla de menor a mayor y visualizarla por pantalla de la forma siguiente:

Tabla Inicial

67
5
23
3
12
4
21
7
88
65

Tabla ordenada

3
4
5
7
12
21
23
65
67
88

37.- Llenar dos tablas de 10 posiciones cada una con números enteros comprendidos entre el 1 y el 99. Ordenar dichas tablas y fusionarlas en una tercera de 20 posiciones, de forma que vaya quedando ordenada de menor a mayor. Ir visualizando el proceso.

Ejemplo

Tabla_1

67
5
23
3
12
4
21
7
88
65

Tabla_2

77
41
53
72
1
2
28
60
63
85

T. Resultado

1
2
3
4
5
7
12
21
23
28
41
53
60
63
65
67
72
77
85
88

38.- Mediante el generador de números aleatorios llenar una tabla de dos dimensiones (10,10) con números enteros comprendidos entre el 1 y el 99. La tabla deberá visualizarse en pantalla así como los números que se van generando, colocándolos en cada casilla de la tabla.

Ordenar la matriz anterior por filas y columnas, de menor a mayor, y visualizar la tabla ordenada a la derecha de la anterior.

Tabla inicial

2	23	34	1	56	4	7	11	22	55
33									
31									

Tabla ordenada

1	2	4	7	11	22	23	34	55	56
31									
33									

CADENAS

39.- Hacer un programa que hace “eco” del flujo de entrada y convierte las palabras en palabras iguales que comienzan con letra mayúscula.

Ejemplo: “cáceres patrimonio de la humanidad”

“Cáceres Patrimonio De La Humanidad”

40.- Hacer un programa que haga “eco” de la entrada poniendo cada palabra en una línea.

41.- Hacer un programa que cuente la ocurrencia de la letra ‘e’ en el flujo de entrada.

42.- Leer 5 cadenas, almacenarlas en un array separadas por blanco y a continuación visualizarla.

43.- Leer una secuencia de nombres, uno por línea, terminado por el valor centinela’\$’. A continuación visualizar los nombres almacenados en el array nombres.

44.- Escribir un programa que devuelve la forma plural de la palabra española que se le pase.

45.- Escribir un programa que lea una secuencia de nombres, uno por línea, los ordene y, a continuación , los presente por pantalla.

46.- Leer una línea de entrada. Descartar todos los símbolos excepto los dígitos. Convertir la cadena de dígitos en un entero y fijar el valor del entero a la variable n.

47.- Mediante el generador de números aleatorios generar un texto (mayúsculas -65 a 90- y minúsculas -97 a 122-) de 250 caracteres. Convertir todo a mayúsculas. Ver cuántas A, B, C...Z hay en el texto.

El texto debe aparecer en 5 líneas de 50 caracteres cada una al principio de la pantalla y a doble espacio. Más abajo aparecerán la tabla siguiente:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	3	0	5	3	1	5	7	8	0	3	1	4	5	7	3	5	7	9	0	1	2	4	6	7	9

La que aparezca mayor número de veces se mostrará en vídeo inverso.

48.- Introducid un texto por teclado (de 250 caracteres). Se debe realizar un programa que cuente el número de preposiciones existentes en dicho texto, listando la frecuencia de cada una de ellas.

El texto debe aparecer en cinco líneas de 50 caracteres cada una al principio de la pantalla y a doble espacio. A continuación se indicará el número total de preposiciones y después una lista de todas las preposiciones y cuántas veces aparece cada una.

ESTRUCTURAS O REGISTROS

49.- Una famosa cadena de televisión desea un programa que le permita conocer el tipo de audiencia que tiene en diferentes franjas horarias. Concretamente hay cinco franjas:

- 1.- Antes de las 14 h.
- 2.- De 14 h. a 16,30 h.
- 3.- De 16,30 h a 20 h.
- 4.- De 20 h a 22 h.
- 5.- Después de 22 h.

En cuanto al tipo de audiencia, distinguimos a los televidentes en función de la edad, estableciendo tres rangos:

- 1.- Menos de 20 años.
- 2.- Entre 20 y 50.
- 3.- Más de 50.

El programa irá recibiendo edad y hora (hora:minutos) sobre personas que ven esta cadena y con esa información tendrá que actualizar la estructura de datos AUDIENCIA que será una matriz de 3 filas y 5 columnas. Las filas representan el tipo de audiencia y las columnas representan las distintas franjas horarias.

Se dispone de la estructura T_hora:

```
struct T_hora{
    int h; // de 0 a 23
    int m; // de 0 a 59
};
```

1.- Se deben codificar las funciones siguientes:

- Inicializar(A:AUDIENCIA). Inicializa la estructura a 0.
- Incluir(A:AUDIENCIA; edad:entera; hora: T_hora). A partir de los datos de un televidente (edad y hora a la que ve la televisión), se modifica AUDIENCIA.
- Consulta(A:AUDIENCIA; franja:entera). Devuelve el número de televidentes de una franja a lo largo de todo el día.
- Máximo(A:AUDIENCIA); Devuelve la franja horaria de más audiencia.

Será necesario codificar además dos funciones auxiliares:

- Conv_hora(h:T_hora). Recibe como dato una hora y calcula la franja horaria a la que pertenece.
- Conv_edad(e:entera). Recibe como dato la edad y calcula el rango al que pertenece.

2.- La función principal deberá:

- Inicializar la estructura.
- Modificar AUDIENCIA según los datos de 10 televidentes.
- Consultar la audiencia de una franja.
- Dar la máxima audiencia.
- Se visualizará la estructura cada vez que se crea necesario.

50- Una empresa se dedica a hacer reformas en pisos y edificios. Para gestionar los proyectos que tienen en marcha han encargado un programa del que se van a implementar algunas funciones y algunas operaciones.

De cada proyecto se guarda un identificador, el nombre del cliente, el número de días aproximados que dura la reforma, y lo que va a costar. El número máximo de proyectos puede ser de 10.

Escribe la estructura de datos más adecuada a utilizar. Utiliza una estructura, un vector y una variable que indica la ocupación del vector.

- Implementa la función Crear que simplemente pone a cero la variable que indica qué ocupación tiene el vector.
- Implementa la función Insertar que introduce los datos de un nuevo proyecto en la estructura. Debe hacerse de tal forma que los proyectos se puedan recuperar conservando el orden de llegada.
- Implementa la función acabado que, dado el identificador de un proyecto, lo marque como finalizado.
- Implementa la función Total que calcula lo que ha ganado la empresa con los proyectos finalizados.
- Implementa la función Primero que devuelve el identificador del proyecto que lleva más tiempo y que no se ha acabado todavía.
- Implementa la función Ajustado que dado un número de días X , devuelva el identificador del proyecto sin finalizar con una duración prevista menor o igual a X , pero lo más cercana posible. (Es decir si el proyecto 1 tiene una duración de 3, el proyecto 2 de 5 y el proyecto 3 de 8, y nos dan un $X = 6$, tendrá que devolver el identificador del proyecto 2).
- Implementa la función Main para probar las funciones anteriores.

FICHEROS DE TEXTO

51.- Haga un programa para mostrar un archivo de texto pantalla a pantalla. Después de completar la pantalla, el programa se para hasta que el usuario pulse una tecla y entonces muestra la siguiente página.

52.- Realizar un programa que cuente las haches intercaladas de un texto de varias líneas que se encuentra en un fichero. Repetir el problema de forma que cuente sólo las palabras con hache intercalada. En este segundo caso una palabra con varias haches intercaladas debe ser contada una sola vez.

NOTA: No son hache intercalada las que son comienzo de palabra ni las que aparecen formando 'ch'.

53.- Haz un programa que copie un archivo en otro carácter a carácter. Los nombres de los archivos origen y destino se pasarán como argumentos.

54.-Crea un programa que cuente las palabras que hay en un archivo de texto.

Nota: escribe una función o una macro que te indique si un carácter es una letra o no. No utilices una función estándar que hay para ello.

55.- Escriba un programa que haga una lista de todas las palabras encontradas en un archivo de entrada y escriba esa lista en un archivo de salida, junto con el número de veces que ha aparecido cada palabra.

56.- Se dispone de dos ficheros de texto que contienen una lista de números enteros cada uno, ordenados de menor a mayor, a razón de un número por línea. Se pide un programa en C que a partir de estos dos ficheros, genere un tercer fichero que contenga todos y cada uno de los datos de los dos ficheros iniciales y esté también ordenado de menor a mayor.

NOTAS: - Cada fichero de datos inicial puede tener un número arbitrario de líneas (y por tanto de números).

Los números de un fichero inicial no tienen ninguna relación con los del otro, siendo por tanto, dos listas de números completamente independientes.

57.- Escriba un programa que lea un fichero de texto con varias líneas y cuente cuantas vocales hay de cada tipo (a,e,i,o,u).

NOTA: Para almacenar el número de vocales usar un array de 5 posiciones.

58.- Se tiene un fichero que contiene dos números enteros en cada línea. Se pide hacer un programa que lea el fichero y realice los siguientes cálculos:

La media aritmética de los números que están en la primera columna

La media ponderada de los números de la primera columna empleando como pesos los de la segunda.

59.- Tenemos un fichero de texto que contiene varias líneas y necesitamos imprimir las iniciales de todas las palabras que aparecen en el mismo. Realizar el análisis y obtener un programa que imprima en pantalla las iniciales separadas por un espacio. Una letra es inicial de palabra si va precedida de blanco o es la primera de una línea. Téngase en cuenta que las palabras del fichero, están separadas por un único espacio en blanco o final de línea.

FICHEROS CON REGISTROS

60.- Se debe guardar un listado de alumnos en un fichero con la siguiente información:

- N° de expediente
- Nombre
- Nota 1° parcial.
- Nota 2° parcial.
- Nota 3° parcial.

El programa debe realizar las siguientes operaciones:

- Crear el archivo pidiendo los datos por consola.
- Obtener un listado en el que aparezcan los datos del estudiante más la media de sus tres notas.
- Obtener un listado con los estudiantes que superen un 7 de nota media.

61.- Creación de un fichero para un sistema de facturación de clientes con la siguiente información por cliente:

- Nombre.
- Dirección.
- N° de cliente.
- Saldo.
- Cuenta (al día, atrasada, deudor). Se debe utilizar un solo carácter para almacenar esta información.

El programa debe realizar las operaciones siguientes:

- Crear el archivo pidiendo los datos por consola.
- Obtener un listado de aquellos clientes que tienen Cuenta=deudor.

62.- Una compañía guarda un fichero de nóminas con los siguientes datos de cada empleado:

- Nombre.
- N° de empleado.
- Nivel (manual, cualificado, oficinista o directivo). Se debe usar un solo carácter para guardar esta información.
- Sueldo bruto.
- Exención de impuesto.
- N° de semanas de antigüedad en la empresa.

Si el nivel es de oficinista o directivo lleva un complemento en el sueldo. Este complemento es una cantidad constante igual para todos.

El fichero se empleará para imprimir los recibos de las nóminas.

El sueldo bruto total que cobran es Sueldo bruto + Complemento.

El impuesto se deduce según la fórmula: $(\text{sueldo bruto total} - \text{exención}) * 0,33$.

Escribir un programa que permita, mediante un menú, realizar las operaciones siguientes:

- Cree el fichero vacío.
- Añada registros.
- Modifique los valores de un registro por el número de empleado.
- Añada una semana de antigüedad a todos los empleados.
- Muestre un listado del siguiente tipo:

<u>Nombre</u>	<u>Nivel</u>	<u>Sueldo</u>	<u>Complemento</u>	<u>Total sueldo bruto</u>	<u>Exención</u>	<u>Impuesto</u>	<u>Sueldo neto</u>
---------------	--------------	---------------	--------------------	---------------------------	-----------------	-----------------	--------------------

63.- Tenemos un fichero de alumnos con sus notas en 4 asignaturas. Se necesita un programa que realice las siguientes operaciones mediante un menú de opciones:

- Cree el fichero y pida los nombres de los alumnos.
- Pida y guarde las notas de los alumnos en cada asignatura, asignatura por asignatura, es decir, debe pedir primero las notas de Matemáticas de todos los alumnos y guardarlas, después las de Lengua y así sucesivamente.
- Mostrar un listado similar al siguiente:

<u>Nombre</u>	<u>Matemáticas</u>	<u>Lengua</u>	<u>Física</u>	<u>Historia</u>	<u>Media</u>
García, J.	6	8	7	8	7.25
Rodríguez, M.	5	4	4	7	5
MEDIAS	5.5	6	5.5	7.5	6.125

64.- Se dispone de un fichero general de alumnos, llamado **ALUMNOS.DAT**, con organización secuencial, que contiene la siguiente información:

Número de expediente
Nombre y apellidos
Dirección
Teléfono
Población
Provincia
Especialidad
Curso (1° a 5°)
Grupo (A ... H)

Se desea efectuar un mantenimiento de dicho fichero, es decir, se procederá a dar en el mismo:

Altas
Bajas
Modificaciones
 Dirección
 Teléfono
 Población
 Provincia
Consultas
Listados

Las cuatro primeras operaciones se realizarán a través del número de expediente. Al dar altas se comprobará que el registro no exista con anterioridad; en bajas, modificaciones (sólo podrán modificarse los campos que se señalan) y consultas si no existe el registro se dará un mensaje indicándolo; en caso de existencia se mostrará por pantalla y se pedirá conformidad de baja (para esta opción) o campo a modificar (en la opción correspondiente). Para los listados se introducirá la especialidad, curso y grupo y se dará un informe de aquellos registros que cumplan estos requisitos, con el formato siguiente:

Nº Exp.	Nombre	Dirección	Tfno.
----------------	---------------	------------------	--------------

Nota.- En las modificaciones podrá ser modificado uno o más campos cada vez.

65.- Se dispone de un fichero directo que contiene la siguiente información:

Descripción
Stock
Stock mínimo
Precio unitario

Se pide realizar un mantenimiento del mismo, es decir, se procederá a dar:

Altas
Bajas
Modificaciones
 Stock
 Stock mínimo
 Precio unitario
Consultas
Listado

El listado se hará de aquellos artículos que se encuentren en Stock mínimo (Stock \leq Stock mínimo), con el formato siguiente:

Código Descripción Stock Stock mínimo

Nota.- El código de artículo, que será consecutivo, indica la posición del registro en el fichero.

66.- Se dispone de un fichero directo (donde el número de representante indica la posición del registro en el fichero, estos se darán de forma consecutiva.), llamado **REPRE**, de los representantes de una empresa, que contiene la siguiente información:

Nombre
Dirección
Total ventas
Importe bruto ventas

Por otra parte tenemos un fichero de movimiento (secuencial), llamado **MOVIM**, que contiene:

Nº de representante
Ventas mensuales
Importe bruto mensual

A) Se desea actualizar el fichero directo, teniendo en cuenta que si las ventas mensuales exceden de 1000 unidades, por cada 500 (por encima de las 1000) se hará una bonificación del 2% sobre el importe bruto mensual, que repercutirá sobre él.

B) Obtener un listado de aquellos representantes que hayan vendido más de 10.000 unidades, dando el importe líquido de las mismas

Nota.- El líquido se obtiene después de aplicar unas retenciones al bruto de la forma siguiente:

Hasta 12.999 euros se retiene un 10%.

A partir de 13.000 las retenciones figuran en una tabla (T) y varían cada 3.000 euros, hasta un tope de 40.000.

A partir de 40.000 la retención es del 35%.

El formato del listado será el siguiente:

Nº Rep. Total ventas Imp. bruto Imp. líquido

67.- Se tiene un fichero maestro (directo, donde el código de cliente indica la posición del registro en el fichero. Los códigos se darán de forma consecutiva.), llamado **USUARIO**, de los usuarios de Cuentas Corrientes de una Caja de Ahorros. Contiene la siguiente información:

Nombre
Dirección
Fecha último movimiento
Saldo

Se tiene a su vez un fichero histórico (secuencial), llamado **HISTOR**, que contiene:

Código de cliente
Tipo de operación (Añadir/Disminuir)
Fecha movimiento
Cuantía movimiento

A) Crear un fichero de movimientos, llamado **MOVIM**, de organización secuencial, que contenga:

Código de cliente
Tipo de operación (A/D)
Fecha movimiento
Cuantía movimiento

B) Actualizar el fichero maestro mediante el de movimiento. Al mismo tiempo cada movimiento de este fichero habrá que reflejarlo en el histórico.

C) Obtener los movimientos de un cliente cuyo código entrará por teclado, desde una fecha introducida también por teclado hasta la actual (esta se cogerá del sistema al principio del programa). El formato del listado será:

Código de cliente	XXXXXXXX	
Nombre	XXXXXXXXXXXXXXXXXX	
Fecha	Tipo	Cantidad
XXXXXX	XXXXXX	XXXXXXXXXX
XXXXXX	XXXXXX	XXXXXXXXXX
XXXXXX	XXXXXX	XXXXXXXXXX
		Saldo final XXXXXXXXXXXX

68.- Se tiene un fichero directo (donde el código de libro indica la posición del registro en el fichero), llamado **LIBROS**, que contiene la siguiente información:

Editorial
Título
Autor
Fecha de salida
Fecha de devolución
Nombre del peticionario
Nº de veces que ha salido

Además disponemos de dos ficheros secuenciales, uno de salidas y otro de devoluciones, con la información siguiente:

SALIDAS

Código de libro
Fecha de salida
Fecha de devolución (Fecha límite)
Nombre del peticionario

DEVOLUCIONES

Código de libro
Fecha de devolución (Fecha real)

A) Crear y dar altas en los tres ficheros.

B) Actualizar LIBROS mediante los ficheros secuenciales.

C) Obtener un listado de los libros cuya fecha de devolución haya expirado, con el formato siguiente:

Título	Peticionario	Fecha dev.
---------------	---------------------	-------------------

Nota. Cada vez que se da una salida se actualizará el campo nº de veces que ha salido.

69.- Queremos informatizar una compañía de seguros de coches; se dispone de tres tipos de seguros:

- A...Obligatorio
- B...Voluntario de daños a terceros
- C...Voluntario a todo riesgo

Por cada tipo de seguro tendremos la siguiente información:

Nombre del asegurado
Marca del vehículo
Fecha de contrato
Precio del coche
Nº de accidentes al año
Nº de años sin accidentes
Cuota a pagar anualmente

A (0.3% precio)
B (1.2% precio)
C (4.0% precio)

Último año cotizado

Un asegurado tendrá normalmente un máximo de dos seguros distintos (A, AB, AC).

El programa consiste en tres ficheros de organización directa (donde el número de póliza indicará la posición en el fichero), uno por cada tipo de seguro, con los campos anteriores. Un fichero secuencial con la información siguiente:

Nombre	
Seguro obligatorio	1 letra 1 nº
Seguro voluntario	1 letra 1 nº

La letra indicará el tipo de seguro (A,B o C) y el número, que será correlativo, el número de póliza, es decir, la posición asociada en el fichero correspondiente.

Tenemos que confeccionar los recibos de los asegurados. Hay un recibo por cada tipo de seguro y se pagan por adelantado.

El formato es el siguiente:

Nombre de la compañía	
Nº de póliza	Nombre
Tipo de seguro	
Periodo de validez de DDMMAAAA a DDMMAAAA	
Marca del vehículo	Cuota anual
Bonificación	Cuota líquida

La bonificación será:

- 1 año sin accidentes 2% cuota anual
- 2 años sin accidentes 3% cuota anual
- 3 ó más años 5% cuota anual

Todos los días 1 de cada mes se sacarán los recibos correspondientes a ese mes.

Cuando se produzca un accidente se pondrá a cero el campo año sin accidente.

Si al confeccionar el recibo el campo de accidentes está a cero y el n° de accidentes año también se sumará uno al campo año sin accidente y se contabilizará para ese recibo.

Cuando el número de accidentes año sea superior a 12 se procederá a dar de baja al asegurado.

Los recibos se sacarán por impresora.

70.- Se dispone de un fichero maestro (**SALDOS.DAT**), de organización secuencial, que contiene la siguiente información:

Código
Nombre
Dirección
Población
Saldo

Por otra parte se tiene un fichero de movimientos (**MOVIM.DAT**), de organización secuencial, el cual además de los campos anteriores tiene otro, llamado Tipo (1 carácter).

Se desea:

Crear el fichero maestro con al menos 10 registros.

Ordenar por código el fichero maestro.

Proceder a dar altas, bajas y modificaciones en el fichero de movimiento de la forma siguiente:

Altas.- el campo Tipo será una "A" y se introducirá toda la información correspondiente.

Bajas.- el campo Tipo será una "B". Se introducirá sólo el código, quedando el resto de los campos en blanco.

Modificaciones.- el campo Tipo será una "M". Se introducirá el código, el campo Nombre se aprovechará para poner una I/R (según se trate de un Ingreso o un Reintegro) y el importe se almacenará en el campo Saldo. El resto de los campos quedará en blanco.

Actualizar el fichero maestro a través del de movimiento, es decir, traspasar las altas, bajas y modificaciones de uno a otro. Para ello lo primero será ordenar el fichero de movimiento. En el caso de encontrarnos con una baja se mostrará el registro por pantalla y se pedirá conformidad.

En el momento de actualizar se pide obtener un listado por impresora de aquellos registros que son alta. Se imprimirá el código, nombre y saldo de cada uno de ellos.

Al finalizar la actualización se borrará el fichero de movimiento.

1) Los ficheros se ordenarán por código.

El maestro se ordenará como último paso de la Creación y el de movimiento como primero de la Actualización.

2) Caso de no existir un registro (bajas o modificaciones), o que ya exista (altas) se sacará por pantalla el mensaje correspondiente.

<p style="text-align: center;">MENÚ PRINCIPAL</p> <p style="text-align: center;">1. MAESTRO 2. MOVIMIENTO 3. SALIDA</p> <p style="text-align: right;">PULSE OPCIÓN :</p>

<p style="text-align: center;">MAESTRO</p> <p>1. CREACIÓN 2. ACTUALIZACIÓN 3. MENÚ PRINCIPAL</p> <p style="text-align: right;">OPCIÓN :</p>
--

<p style="text-align: center;">MOVIMIENTO</p> <p>1. ALTAS 2. BAJAS 3. MODIFICACIONES 4. MENÚ PRINCIPAL</p> <p style="text-align: right;">OPCIÓN:</p>

71.- Sea el fichero **FICH1.DAT**, de organización secuencial, en el que figuran los siguientes datos:

Código cliente
Saldo anterior
Saldo actual
Fecha último movimiento

En otro fichero (**FICH2.DAT**), de organización secuencial, están los datos siguientes:

Código cliente
Nombre
Dirección
Localidad

1. Obtener un fichero conjunto (**FICH3.DAT**), con organización directa, donde la posición del registro en el fichero viene indicada por el código de cliente. Ambos ficheros deberán ser ordenados por dicho campo. Aquellos registros que sólo aparezcan en uno de los ficheros se ignorarán al formar **FICH3**.

2. Dar por impresora un listado de aquellos clientes cuyos saldos sean negativos. Por cada mes que pase desde la fecha del último movimiento se aplicará un 5% de interés. Al final del informe dar el total de la suma de saldos negativos y de los intereses.

<u>C. cliente</u>	<u>Nombre</u>	<u>Saldo anterior</u>	<u>Saldo actual</u>
XXX	XXXXX	XXXXXX	XXXXXX
XXX	XXXXX	XXXXXX	XXXXXX
XXX	XXXXX	XXXXXX	XXXXXX

Total saldos : XXXXXXXXXXX

Total intereses : XXXXXXXXXXX

72.- Se han obtenido una serie de datos para una encuesta. Están en un fichero con organización secuencial, llamado **ENCUESTA**, y contiene:

Código concepto

Clave de edad

Respuesta

0..si

1..no

2..ns/nc

Las claves de edad son:

A....entre 20 y 30 años

B....entre 31 y 40 años

C....entre 41 y 50 años

D....entre 51 y 60 años

E....a partir de 60 años

Existe por otro lado un fichero secuencial llamado **CONCEP**, que contiene la siguiente información:

Código de concepto

Denominación

Obtener un listado por conceptos con el formato siguiente:

DENOMINACIÓN: XXXXXXXXXXXXXXXXXXXXX

	SI	%SI	NO	%NO	NS/C	%NS/C
20/30	XXX	XXX	XXX	XXX	XXX	XXX
31/40	XXX	XXX	XXX	XXX	XXX	XXX
41/50	XXX	XXX	XXX	XXX	XXX	XXX
51/60	XXX	XXX	XXX	XXX	XXX	XXX
más 60	XXX	XXX	XXX	XXX	XXX	XXX

DENOMINACIÓN: XXXXXXXXXXXXXXXXXXXXX (siguiente concepto).

73.- Se dispone de un fichero secuencial de habitantes, que contiene la siguiente información:

Código de habitante
Nombre y Apellidos
Dirección
Categoría de la calle
Piso
Número de balcones

Se desea:

A) Crear y dar altas en el fichero.

B) Obtener un listado que indique lo que debe pagar cada habitante que tenga algún balcón.

Se tiene una tabla de dos dimensiones (10 filas y 8 columnas), donde la fila indica la categoría y la columna el piso. Si el piso fuese superior a 8 se tomará en la tabla la posición correspondiente al 8.

Esta tabla se cargará a principio de programa con cantidades fijas. Cada una de estas cantidades será la cuota correspondiente a categoría y piso, que deberá multiplicarse por el número de balcones.

LISTADO DE RESULTADOS

Código	Nombre	Categoría	Piso	Nº Balc.	Cuota	Total
--------	--------	-----------	------	----------	-------	-------

74.- Se desea informatizar un hotel. Para ello disponemos de los ficheros siguientes:

1. Fichero de movimiento de habitaciones (directo por nº de habitación)

Tipo (I, D, T)
Fecha de entrada
Nombre del cliente
Código cliente
Dirección
Servicios
 Comedor
 Cafetería
 Teléfono
 Lavandería
 Varios

2. Fichero histórico de habitaciones (Secuencial)

Número de habitación
Tipo (I, D, T)
Fecha de entrada
Código de cliente
Días de estancia
Importe por habitación
Importe por servicios

3. Fichero histórico de clientes (directo, por código de cliente)

Nombre
Dirección
Días de estancia mes actual
Importe
Días de estancia total
Importe

Nota.- El importe de las habitaciones por día es:

Individual	50 euros.
Doble	70 euros.
Triple	90 euros.

Tanto el importe por habitación como por servicios lleva un 7 por ciento de I.V.A.

Al comienzo del programa se cargará una tabla con todos los números de las habitaciones de que dispone el hotel (tanto libres como ocupadas). También se debe introducir la fecha actual (se cogerá del sistema).

Las operaciones que se realizan en el hotel son las siguientes:

ENTRADAS (Cuando un cliente ingresa en el hotel).

Se dará de alta en el fichero de movimiento de habitaciones, comprobando la existencia o no de la habitación, así como la posibilidad de que esté ocupada.

SERVICIOS (Cada vez que un cliente haga uso de cualquiera de los servicios de que dispone el hotel).

Se actualizará el importe de los mismos por cada concepto en el fichero de movimiento de habitaciones.

SALIDAS (Cuando un cliente abandona el hotel).

Se procederá a dar el alta correspondiente en el fichero histórico de habitaciones, así mismo se actualizará o se dará de alta (dependiendo de si existe o no con anterioridad) en el fichero histórico de clientes. Por último se dará la baja pertinente en el fichero de movimiento de habitaciones.

LISTADO MENSUAL (Se dará de forma automática el último día del mes). Se realizarán las operaciones siguientes:

Alta en el fichero histórico de habitaciones.

Actualización del histórico de clientes.

Actualización del de movimiento de habitaciones (se pondrá como fecha de entrada el día 1 del próximo mes y los servicios a cero).

Los pasos anteriores se realizarán de manera conjunta.

Ordenación del histórico de habitaciones, por número de habitación.

Listado con el formato siguiente:

Nº hab.	Tipo	Días ocup.	Imp. hab.	Imp. ser.	% ocup.
XXX	X	XXX	XXXXXX	XXXXXX	XX%
XXX	X	XXX	XXXXXX	XXXXXX	XX%
Importe de todas las habitaciones			XXXXXXXX		
I.V.A. de todas las habitaciones			XXXXXX		
Importe de todos los servicios			XXXXXXXX		
I.V.A.			XXXXXX		
Total importes			XXXXXXXXXX		
Total I.V.A.			XXXXXXXXXX		

CLIENTE DEL MES

Nombre	Dirección	Días	Importe
XXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXX	XX	XXXXXX

Actualización histórico de clientes (se procederá a pasar los días de estancia del mes actual al total, así como el importe; dejando esos campos a cero).

75.- Para la informatización de un Instituto se dispone de los ficheros indicados a continuación con la información siguiente:

Fichero directo de alumnos (por nº de expediente)

Nombre del alumno
Tabla de notas
Notas

Las notas figuran en una tabla, habiendo un máximo de 13 asignaturas. Esta tabla está inicializada con el valor -1.

Fichero directo de asignaturas (por código de curso)

Tabla
Nombre de la asignatura
Nº alumnos que la cursan

La tabla tiene dos posiciones, en la primera figuran los nombres de las diferentes asignaturas del curso (13 como máximo). La segunda posición se encuentra inicializada a cero.

Fichero secuencial de movimiento

Código de curso
Número de expediente
Nombre de la asignatura
Nota

Se pide:

A) Actualizar los ficheros directos a través del de movimiento.

1. Se incrementará en uno la posición correspondiente de la tabla del fichero de asignaturas, obteniendo así el número total de alumnos que cursan cada una de ellas y la posición de la tabla de notas del fichero **ALUMNOS** para el punto 2.
2. Se pasará la nota de la asignatura a la posición que corresponda en la tabla de notas del fichero de alumnos.

B) Realizar el siguiente listado:

CURSO : XXXXX

EXP.	NOMBRE	NOTA MEDIA
XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XX
XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XX

MEDIA POR ASIGNATURA

ASIGNATURA	NOTA MEDIA
XXXXXXXXXXXXXXXXXX	XX
XXXXXXXXXXXXXXXXXX	XX
.	
.	
.	
(Siguiete curso...)	

Nota:

- 1) Al realizar los listados se comprobarán las notas, si alguna fuese -1 indicará que el alumno no cursa esa asignatura, y por lo tanto, no se tendrá en cuenta para la media del alumno ni para la media por asignatura.
- 2) Tened en cuenta, al realizar la media por asignatura, que en la tabla del fichero indexado de asignaturas tenéis el número de alumnos que cursan cada una de ellas.

PUNTEROS Y ESTRUCTURAS DINÁMICAS

76.- Escribe un programa que cree un array dinámico de números reales. Pidiendo el número de elementos y el valor de cada elemento. Después deberá normalizar el vector y mostrarlo en pantalla.

Nota: normalizar un vector es dividir todos los valores de sus elementos por el mayor, de modo que todos queden comprendidos entre 0 y 1.

77.- Realiza un programa que cree una matriz dinámica de números enteros, pidiendo previamente el número de filas y columnas. Rellena la matriz pidiendo los valores de cada elemento al usuario y por último busca en la matriz el elemento mayor y el menor, mostrándolos en pantalla junto con su posición, es decir, su fila y su columna.

78.- Escribe un programa que utilice un array dinámico para guardar varias estructuras del tipo:

```
struct dato
{
    char nombre [20];
    int edad;
};
```

el programa tendrá cuatro funciones para realizar las siguientes operaciones:

- a- Crear el array pidiendo el número de elementos que debe tener y los datos de cada uno.
- b- Ordenar el array por el nombre.
- c- Ordenar el array por la edad.
- d- Listar los datos por pantalla.

79.- Multiplicar dos matrices de dimensiones cualesquiera de números enteros. (Utilizar matrices dinámicas). (Se debe cumplir que $\text{columnas}_1 = \text{filas}_2$ para poder multiplicar dos matrices).

80.- Un polinomio puede ser representado por una lista en la que cada nodo contiene un coeficiente, un exponente y un puntero o enlace al siguiente elemento. Escribe un programa que ejecute la suma y la diferencia de dos polinomios almacenados de esta forma.

81.- En un archivo de texto figura en cada línea un número entero positivo arbitrariamente grande, de forma que no puede ser almacenado en ningún tipo de dato predefinido. Diseñar un programa que sume todos los números del archivo y muestre el resultado final, así como los números de cada línea.

82.- Dadas dos listas dinámicas que almacenan números enteros, escribir mediante un programa todas las combinaciones posibles de parejas de números formadas por un número de la primera lista y otro de la segunda, cuya suma sea menor o igual que un número N introducido por teclado.

83.- Diseñar una función recursiva para mostrar los datos de una lista enlazada, y otra para listarla invertida.

84.- Dado que en las colas la inserción se produce siempre por el final, diseñar un programa que gestione una cola con dos punteros externos: un puntero al principio de la cola y otro al final de la misma.

85.- Una empresa dispone de un fichero de almacén. Este fichero está guardado en disco y contiene la siguiente información de cada producto:

- Nombre del producto.
- Código.
- Precio.
- Stock.

Hacer un programa que mediante un menú realice las siguientes operaciones:

- i) Crear el fichero vacío.
- ii) Añadir un nuevo producto.
- iii) Mostrar todos los productos ordenados por el precio (descendente).
- iv) Mostrar todos los productos ordenados por el nombre (ascendente).

Para hacer esto el programa debe mantener dos índices en sendas listas dinámicas ordenadas respectivamente por el precio y por el nombre, con la posición de cada registro dentro del fichero.

Estas listas se deben generar al iniciarse el programa con la información de los registros existentes en el fichero, y deben mantenerse actualizadas al añadir nuevos registros o al crear el fichero vacío.

86.- Se tienen dos listas dinámicas de números enteros ordenadas, escribe una función que cree una tercera lista ordenada con todos los elementos de las dos primeras.

87.- Se tiene una lista ordenada de números enteros. Escribe una función que genere otra nueva lista con los mismos elementos de la anterior pero en orden inverso.

88.- Escribe una función que busque en una lista dinámica de números enteros una pareja (x, y) de números de esa lista, tal que la expresión $(x^2 - 10y)$ nos dé un valor mínimo. Los números x e y pueden ser el mismo.

89.- Escribe una función que busque en una lista dinámica de números enteros una pareja (x, y) de números de esa lista de forma que x sea un número par e y sea un impar.

PROGRAMACIÓN ORIENTADA A OBJETO

90.- Crear una clase “círculo”: en su construcción debe recibir el valor del radio, y debe tener funciones miembro que permitan calcular la longitud de la circunferencia y el área del círculo. Usarla en un programa para calcular longitudes y el área de algunos círculos.

91.- Usando como clase base la del ejercicio 1º, crear una nueva clase "cilindro" que en su creación reciba el radio y la altura y disponga de funciones para hallar su superficie y su volumen.

92.- Usarla en un primer programa para hallar el volumen y la superficie de un cilindro.

93.- Usarla en un segundo programa que pida los datos de dos cilindros y calcule la suma de sus superficies y volúmenes.

94.- Diseña un programa para calcular la hipotenusa y los ángulos agudos de un triángulo rectángulo dados los catetos. Usar una clase triángulo que reciba como parámetros en su construcción los valores de los catetos y disponga de funciones miembro los ángulos y la hipotenusa.

95.- Hacer un programa que sume, reste, multiplique y divida fracciones, también deberá ser capaz de simplificar una fracción. Usar una clase fracción que guarde el numerador y denominador de una fracción y permita realizar todas estas operaciones con funciones miembro que reciban como parámetro si es necesario otros objetos de tipo fracción, por ejemplo: la función suma deberá recibir dos parámetros de tipo fracción que serán los sumandos y el resultado se guardará en el propio objeto fracción que llama a dicha función.

96.- Definir una clase "fecha" para guardar fechas. Con un constructor por defecto que cree la fecha 1-1-1900, otro constructor que cree el objeto con la fecha pasada por parámetros que debe comprobar la validez de dicha fecha, si es errónea guardará la fecha por defecto; además debe contar con las siguientes funciones miembro:

- Mostrar, que mostrará la fecha en pantalla.
- int anno(), que devolverá el valor del año.
- int mes(), que devolverá el valor del mes.
- int dia(), que devolverá el valor del día.

Escribe un programa que utilice esta clase para leer varias fechas y las escriba después en pantalla.

97.- Diseña una clase que contenga un array dinámico de números enteros con los siguientes miembros:

- Un constructor que reserve la memoria suficiente para el n° de elementos que se le pasa como parámetro y ponga todos los elementos a cero.
- Dos métodos que permitan respectivamente conocer y modificar el valor de un elemento del array pasándole su posición como parámetro.
- Método que devuelva el número de elementos del array.
- Método que ordene el array.
- Un destructor.

Utiliza esta clase en un programa que cree un array, le asigne valores a los elementos y después los muestre ordenados en pantalla.

98.- Crea una clase derivada de la clase "array" del ejercicio anterior, para aumentar su funcionalidad, permitiendo dos operaciones nuevas:

- Añadir un elemento al array al final del mismo con el valor del parámetro que se le pase a la función.
- Eliminar un elemento del array, indicándole la posición mediante un parámetro.

Nota: en la clase base array los datos privados se deben convertir en protegidos.

Escribe un programa que utilice esta nueva clase array dinámico creando al principio el objeto array con cero elementos y a continuación muestre un menú que permita añadir un elemento, borrar un elemento, ordenar el array y listar los datos del array.