Universidad de Valladolid
Departamento de Informática
Escuela Técnica Superior de Ingeniería Informática   Campus Miguel Delibes, s/n. 47011 Valladolid   Tel.:(983) 423660 Fax:(983) 423816

Programación I

Ingeniería Técnica Informática

Ejercicios de los Temas 10, 11 y 12

Arrays, Registros, Strings y Conjuntos

76.- Elaborar un subprograma para normalizar un vector, con el criterio de que el nuevo vector resultante tenga módulo 1.

77.- Elaborar un subprograma para normalizar un vector, con el criterio de que el nuevo vector resultante tenga como mayor componente una con módulo 1.

78.- Escribir un subprograma que lea una matriz de tamaño $m * n$ y escriba la suma de los elementos de cada fila y la suma de los elementos de cada columna.

79.- (Examen curso 86/87). Dada una matriz de orden $m * n$, escribir un programa que escriba la matriz que resulta de restar a cada elemento la media aritmética de su columna.

80.- Elaborar un subprograma que almacene en una matriz M una serie de datos numéricos y escriba el resultado de restar a cada elemento $M[i, j]$ la media aritmética de la media aritmética de la fila i y la media aritmética de la columna j.

Ejemplo:

ENTRADA: $\begin{vmatrix} 2 & 5 & 8 \\ 3 & 9 & 6 \\ 7 & 5 & 0 \\ 8 & 5 & 2 \end{vmatrix}$ SALIDA: $\begin{vmatrix} -3{,}0 & -0{,}5 & 3{,}5 \\ -2{,}5 & 3{,}0 & 1{,}0 \\ 2{,}5 & 0{,}0 & -4{,}0 \\ 2{,}5 & 0{,}0 & -4{,}0 \\ 3{,}0 & -0{,}5 & -2{,}5 \end{vmatrix}$

81.- (Examen curso 86/87). Sea A un vector de 10 números reales no nulos. Se desea dividir todos sus elementos por el elemento que ocupa la posición k. ¿Es correcto el siguiente método para conseguirlo?

```
Var
        j,k : integer;
        A : array[1..10] of real;
begin
        (* leer k *)
        ............
        (* leer A *)
        ............
        for j:=1 to 10 do A[j] := A[j]/A[k];
        ............
        ( * escribir A *)....
end.
```

82.- Para operar con números enteros de tamaño grande se puede utilizar un vector cuyas componentes sean dígitos decimales (entre 0 y 9, ambos inclusive). Elaborar un subprograma Pascal para sumar números enteros de hasta cuarenta cifras por el procedimiento anterior.

83.- Supongamos que una determinada colección consta de 225 cromos que se venden en sobres, conteniendo cada uno 6 cromos distintos. Elaborar un programa que simule la compra del sobre y nos indique el número de sobres necesarios para completar la colección.

84.- (Examen curso 99/00) Dada como entrada una matriz de caracteres de m filas y n columnas construir un subprograma que escriba en pantalla las cadenas de caracteres de dicha matriz formadas por las diagonales, según se observa en el siguiente ejemplo:

ENTRADA: $\begin{vmatrix} a & b & c \\ d & e & f \end{vmatrix}$ SALIDA:
c
bf
ae
d

85.- Dada una matriz cuadrada, se dice que posee simetría central si el valor de cada elemento coincide con el de su simétrico (ver figuras). Dada una matriz M definida como M : array[1..n,1..n] of tipo, donde tipo está definido en la sección type y n en la sección const (n es entera y puede ser par o impar), escribir un subprograma que compruebe si M posee simetría central

```
x-----
--y---
------
------
---y--
-----x
```

```
a--c-
q----
--o--
----q
-c--a
```

86.- (Examen curso 90/91) Mostrar de qué forma puede representarse un conjunto de números enteros de valor comprendido entre 1 y n mediante un vector de componentes booleanas. Escribir después un subprograma que lea dos de esos conjuntos y calcule su unión y su intersección.

87.- Se dispone de un vector, parcialmente ocupado, que almacena números enteros clasificados en orden creciente. Se desea conocer si un número introducido por teclado se encuentra en el vector, en caso afirmativo se mostrará su posición, y en caso negativo se insertará en la posición que le corresponda. Si se intenta insertar y el vector está lleno, se mostrará un mensaje de error.

88.- (Examen curso 94/95) Para analizar una imagen de una porción de cielo se obtiene una matriz de 40x20 componentes enteras, cada una de las cuales indica la intensidad de luz recibida en un punto de la imagen, entre 0 y

Universidad de Valladolid
Departamento de Informática

Escuela Técnica Superior de Ingeniería Informática — Campus Miguel Delibes, s/n. 47011 Valladolid — Tel.:(983) 423660 Fax:(983) 423816

100. Se supone la existencia de una estrella en una componente de la matriz cuando la suma de las intensidades correspondientes a la misma componente más la media de los 8 puntos circundantes sea mayor o igual que 60. (Se supone que en el borde de la matriz no existen estrellas) Elaborar un programa que lea por teclado una matriz de intensidades y muestre en pantalla la imagen resultante. Las estrellas se representarán por el carácter * mientras que los puntos en que no aparezcan se señalarán mediante un espacio en blanco.

89.- (Examen curso 88/89). El juego de la batalla naval se desarrolla sobre una matriz de 10x10 casillas. Construir un programa que permita al ordenador jugar el papel pasivo. Se jugará con un solo barco de cinco casillas. El programa debe:

- Poner el barco en la matriz.
- Permitir que el adversario tire sobre las casillas.
- Evaluar los incidentes eventuales e informar de ello al adversario.

La cuadrícula donde se visualizará el estado de la partida será:

	0	1	2	3	4	5	6	7	8	9
A	.	.	.	.	.	.	.	.	.	.
B	.	.	.	.	.	.	.	.	.	.
C	.	.	.	.	.	.	.	.	.	.
D	.	.	.	.	.	.	.	.	.	.
E	.	.	.	.	.	.	.	.	.	.
F	.	.	.	.	.	.	.	.	.	.
G	.	.	.	.	.	.	.	.	.	.
H	.	.	.	.	.	.	.	.	.	.
I	.	.	.	.	.	.	.	.	.	.
J	.	.	.	.	.	.	.	.	.	.

90.- (Examen curso 89/90) Sean dos vectores cuyas componentes son de tipo enumerado definido por el usuario. Diremos que a <b si:

- Existe una posición, i, donde un elemento de a es menor que el elemento correspondiente de b ($a_i < b_i$), siendo los elementos anteriores iguales.
- O bien, siendo todos los elementos de a iguales a los elementos correspondientes de b, el número de elementos de a sea menor que el de b.

Elaborar un algoritmo que devuelva -1 , 0 ó 1 dependiendo de si a <b, a = b o a >b.

91.- (Examen curso 91/92) Una partícula realiza una marcha aleatoria conforme a las siguientes reglas: En t = 0 la partícula se encuentra en el origen x = 0, y = 0. En los tiempos t = 1, 2, 3,... la partícula se desplaza a una cualquiera de las cuatro posibles direcciones dadas por $x \to x+1$, $y \to y+1$, $x \to x-1$ y $y \to y-1$. La caminata termina cuando $X^2 + Y^2 \geq R^2$

Elaborar un programa que nos presente en pantalla una tabla con diez valores de R espaciados proporcionalmente (desde un valor mínimo a uno máximo) y los diez tiempos medios que resultan de realizar N caminatas para cada uno de los diez valores de R.La entrada que necesitará el programa serán los valores mínimo y máximo de R, y el número de repeticiones N.

Nota: Para generar valores aleatorios se puede utilizar la función predefinida random.

92.- (Examen curso 95/96) Se dispone de un conjunto variable de números enteros, que se encuentran almacenados en la siguiente estructura:

```
Const MaxNums = ...;
Type TVector = Array[0..MaxNums] of Integer;
```

Donde la componente cero no almacena datos válidos, sino que indica el número de valores almacenados en el vector (por lo tanto, los datos válidos en el vector comprenden desde la componente 1 hasta la componente con ése índice).

Diseñar un procedimiento, inspirado en alguno de los métodos de ordenación por inserción, que nos ordene el vector y además nos elimine los valores duplicados. Razonar sobre la idoneidad del método de ordenación elegido.

Nota: El proceso de eliminación de duplicados debe realizarse simultáneamente a la ordenación del vector, no como un proceso aparte.

93.- (Examen curso 98/99) En el juego del ajedrez, el alfil es una pieza que se puede mover diagonalmente cualquier numero de casillas a lo largo del tablero. Se pide escribir un subprograma que indique si, dada una determinada disposición de las piezas en un tablero, es posible mover un alfil desde su posición actual (fila y columna inicial) a otra (fila y columna final). El subprograma recibe como parámetros la disposición de las piezas en el tablero en una variable de tipo TTablero, y cuatro números enteros en el rango de 1 a 8 que indican la fila y columna inicial y la fila y columna final.

La declaración del tipo TTablero es la siguiente:

```
TCasilla = (Vacia,Peon,Caballo,Alfil,Torre,Reina,Rey);
TTablero = Array[1..8,1..8] of TCasilla;
```

El subprograma deberá devolver como resultado un valor de tipo boolean, true si se puede realizar el movimiento y false si no es posible. El movimiento se puede realizar si se dan las siguientes condiciones:

- En la posición inicial se encuentra efectivamente un alfil en el tablero.
- El movimiento especificado por las posiciones inicial y final es diagonal.
- No existen piezas en la trayectoria seguida por el alfil (no se contempla el caso de que el alfil “coma” a otra pieza).

94.- (Examen curso 98/99) Para representar números reales positivos se ha definido el siguiente tipo de datos:

```
Const
  MaxDig = ..;
```

```
Type
  TNumero = Record
              Digitos : Array[1..MaxDig] of 0..9;
              PosicPtoDecimal : 1..MaxDig
            End;
```

El campo Digitos almacena secuencialmente los dígitos del número representado, alineados a la derecha y añadiendo los ceros a la izquierda necesarios para obtener un total de MaxDig dígitos. El campo PosicPtoDecimal sirve para separar la parte entera del número de la parte decimal. Por ejemplo, suponiendo que MaxDig tuviera el valor 20, el valor 3.141592 se almacenaría de la siguiente forma en una variable de tipo TNumero:

- Campo Digitos:

0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	4	1	5	9	2

- Campo PosicPtoDecimal: 14

Por lo tanto, los dígitos del rango 1..PosicPtoDecimal representan la parte entera, y el resto del array (PosicPtoDecimal+1..MaxDig) la parte decimal. Es interesante notar que si PosicPtoDecimal = MaxDig el número no tiene parte decimal. Utilizando esta estructura, y suponiendo que no se van a producir situaciones de desbordamiento o pérdida de precisión, obtener:

- Un procedimiento que escriba un valor de tipo TNumero en un dispositivo de tipo texto ocupando un determinado número de caracteres (NumCar) y con una precisión (NumDec, número de dígitos decimales) determinada. La cabecera del procedimiento sería la siguiente:

  ```
  procedure Escribir(var Disp:Text;Valor:TNumero;NumCar,NumDec:integer);
  ```

  El procedimiento usará las mismas reglas que la salida formateada de la sentencia write: Se añaden los espacios en blanco iniciales que sea preciso para que la salida tenga NumCar caracteres, y si el número ocupa más caracteres que los especificados en NumCar, se ignora NumCar y se escribe la totalidad del número (sin añadir espacios en blanco).
  Por ejemplo, si la variable N de tipo TNumero almacena el valor 3.141592, se producirían los siguientes resultados ('·' representa un espacio en blanco):
  Escribir(Output,N,10,3) ->·····3.141
  Escribir(Output,N,2,7) ->3.1415920
- Un subprograma que devuelva el valor de tipo real que más se aproxima al que se almacena en una variable de tipo TNumero.
- Un subprograma que devuelva el resultado de la suma de dos valores TNumero en una variable de tipo TNumero.

95.- (Examen curso 86/87) Elaborar la función QuitaBlancos que devuelva la cadena de caracteres resultante de eliminar todos los blancos de la cadena de caracteres que recibe como parámetro. Ejemplo: QuitaBlancos('hoy es 22') debería devolver 'hoyes22'.

96.- (Examen curso 86/87). Dada la siguiente definición de tipo, Tfecha, construir la función edad que devuelva un entero indicando la edad actual en años de una persona. La fecha de nacimiento y la fecha actual se pasan como parámetros de tipo TFecha a la función:

```
TFecha = Record
   Dia  : 1..31;
   Mes  : 1..12;
   Anno : Integer
End;
```

97.- Elaborar un subprograma que nos muestre por pantalla la lista de letras que no aparecen en la cadena de caracteres que se le pasa por parámetro.

98.- (Examen curso 96/97) Escribir una función que devuelva una cadena conteniendo la traducción de un numero decimal (que recibe como parámetro) a número romano. La función deberá verificar que el parámetro es mayor que cero y menor que tres mil, en caso contrario devolverá como resultado la cadena “error de rango”.

Indicación: En numeración romana, los símbolos usados son: I - 1, V - 5, X -10, L - 50, C - 100, D - 500, M - 1000. Un símbolo de valor menor situado a la izquierda de uno de valor mayor tiene el significado de resta. Si está situado a la derecha, tiene el significado de suma.

99.- (Examen curso 96/97) Escribir un procedimiento, Ajustar, que elimine todos los caracteres espacio que aparezcan al principio y final de la cadena (definida como TCadena = String[80]) que se le pasa por parámetro, excepto si la cadena está constituida únicamente por caracteres espacio. Por ejemplo, si el parámetro contiene la cadena “···Esto·es·un··ejemplo··”, despues de la llamada al procedimiento contendrá la cadena “Esto·es·un··ejemplo”

100.- Elaborar un programa para calcular, dados dos valores positivos N y M introducidos por teclado y menores de 256:

- Los divisores de N.Los divisores de M
- Los números que dividen a N, M o ambos.
- Los números que dividen a N y M
- Los números que dividen a N pero no a M
- Los números menores que M que no dividen a N.

101.- Sea A el conjunto de los divisores de 36, B el de los de 60, y C el de los de 90. Calcular mediante un programa Pascal

- A, B y C.
- Divisores comunes a 36, 60 y 90.
- Los divisores de 36 que no son divisores de 60 y 90.

102.- (Examen curso 87/88). En mecanografía cada uno de los ocho dedos cubre un rango de teclas tal como se muestra a continuación :

Dedos mano izquierda	Teclas	Dedos mano derecha	Teclas
2	RTFGVB	2	YUHJNM
3	EDC	3	IK
4	WSX	4	OL
5	QAZ	5	P

Para comprobar la habilidad mecanográfica es útil tener frases que impliquen la utilización de todos los dedos al menos una vez (Por ejemplo, "Pan con pan es comida de tontos"). Escribir un programa que lea frases por teclado, hasta que se introduzca una linea vacía, e imprima la puntuación global usando el siguiente baremo:

- 1 punto por cada frase que implique la utilización de todos los dedos.
- -0.25 puntos por cada frase que no requiera todos los dedos.

103.- (Examen curso 96/97) El juego del fictionary se desarrolla sobre un tablero con 64 casillas, numeradas del 1 al 64. Participan 4 jugadores, cada uno de los cuales dispone de cinco fichas, indistinguibles entre sí. Los jugadores y sus fichas se identifican por los colores rojo, amarillo, azul y verde. En un momento cualquiera del juego, todos los jugadores tienen todas sus fichas repartidas por las casillas del tablero, pero con la restricción de que todas las fichas de un jugador deben de estar en casillas distintas. Sin embargo, en una misma casilla pueden existir fichas de jugadores distintos. A la hora de desarrollar una aplicación para simular este juego, tres programadores han desarrollado tres estructuras de datos distintas. Estas estructuras deben almacenar la información necesaria para conocer en que parte del tablero se encuentra cada una de las fichas de cada jugador en un momento dado.

Definiciones comunes a las tres soluciones

```
Tjugador = (amarillo,rojo,verde,azul)
TnumFicha = 1..5;
TnumCasilla = 1..64;
```

Solución 1- para cada casilla del tablero se almacena la información de los jugadores que tienen ficha en ella

```
Tcasilla = set of Tjugador;
Tinfo = array [TnumCasilla] of Tcasilla;
```

Solución 2- para cada ficha de cada jugador se almacena la casilla donde está

```
Tinfo = array [Tjugador, TnumFicha] of TNumcasilla;
```

Solución 3- para cada jugador se almacenan el conjunto de casillas donde están sus fichas

```
TFichas = set of Tnumcasilla;
Tinfo = array [Tjugador] of TFichas;
```

Se pide desarrollar en Pascal, para cada una de las soluciones, los subprogramas siguientes:

- Escribir una función que devuelva un valor lógico indicando si en una determinada casilla existe una ficha de un determinado jugador.
- Escribir un procedimiento que escriba por pantalla el número de las casillas donde un determinado jugador tiene sus fichas.
- Escribir, para las soluciones en que pueda ser necesario, una función que devuelva un valor lógico resultado de la verificación de que todos los jugadores tienen exactamente cinco fichas (ni una mas ni menos) colocadas en el tablero.

Suponiendo que se elimina la restricción de que en cada casilla sólo pueda existir una ficha de cada jugador, indicar cuales de las estructuras anteriores serían válidas, y por qué.

Problemas Adicionales (ACM 1999)

104.- Anagramas. Se dice que una palabra es un anagrama de otra cuando contienen las mismas letras y el mismo número de veces cada una, aunque en orden distinto. Por ejemplo 'sacos', 'cosas' y 'casos' son todas ellas anagramas. Análogamente, se dice que dos palabras son locogramas si contienen las mismas letras, incluso si alguna de ellas se encuentra repetida. Así, 'cursar' y 'surcas' son locogramas, aunque no son anagramas. Sin embargo 'cursar' y 'cura' no son ni locogramas ni anagramas. Crear un programa que lea dos palabras, de 20 caracteres como máximo, y las clasifique como: anagramas, locogramas o diferentes. Nivel de dificultad: Fácil. Elementos de Programación: Conjuntos, Vectores y Strings.

105.- El Efecto 2000. Existen rumores de que muchos ordenadores van a tener problemas con el año 2000. En algunos equipos que sólo utilizan dos dígitos para representar el año, pasarán bruscamente del año 1999 al año 1900. Otros equipos utilizan un entero de 32-bits para almacenar el número de segundos transcurridos desde una fecha inicial, y en ellos el problema sucederá cuando hayan transcurrido 232 segundos desde esa fecha (unos 136 años), momento en el volverán a marcar la fecha inicial. El problema se puede generalizar de la siguiente forma: Para un determinado ordenador (Oi), existe un año (Aerror) que no puede representar, y cuando su fecha interna llegue a ese año, en su lugar pasará a indicar un determinado año inicial (Ainicial). Un administrador que tiene a su cargo varios ordenadores con distintas versiones del error anterior, decide crear un programa que lea el año que marca cada uno de los ordenadores y, conociendo la versión del error de cada uno de ellos, calcule cual es el verdadero año en que estamos, sabiendo que ese año debe ser superior a 1937 (Nota: 1937 es la fecha en que entro en servicio el primer ordenador, el Z1 de Konrad Zuse). Por ejemplo, si dispone de tres ordenadores con las siguientes versiones del error:

	Aerror	Ainicial	Año mostrado
O1	2012	1979	2009
O2	2000	1900	1941
O3	2040	1904	2005

Se puede comprobar, haciendo una lista de los posibles años reales para los cuales el ordenador mostraría el mismo valor (debido a su fallo), que el menor año compatible con la fecha que muestran los tres ordenadores es 2.141, por lo que el programa decidiría que 2.141 es el año en que estamos.

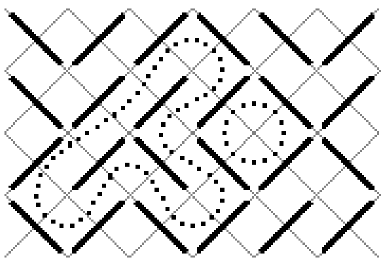
	Año mostrado	Años compatibles con la fecha mostrada					
O1	2009	2009	2042	2075	2108	2141	2174
O2	1941	1941	2041	2141	2241	2341	2441
O3	2005	2005	2141	2277	2413	2549	2685

Se pide crear un programa que solicite el número de ordenadores implicados (menor que 20), y que lea a continuación, para cada uno de ellos, los valores del año que no puede representar (Aerror), el año que muestra en su lugar (Ainicial), y el año mostrado, y que proporcione como salida el primer año compatible con lo que muestran todos los ordenadores (el año real), suponiendo que es menor de 10.000. Si ningún año menor de 10.000 fuera compatible, se mostraría un mensaje de error. Nivel de dificultad: Asequible. Elementos de Programación: Arrays de Conjuntos.

106.- El Entrenador Rebotado. El seleccionador de un equipo nacional de fútbol se encuentra molesto con los comentarios adversos que recibe de una cadena de televisión, pero no puede desahogarse porque el presidente de la federación le ha prohibido hacer declaraciones a la prensa. En su lugar, a ideado una venganza más sutil: Ordenar a sus jugadores por altura mediante una disposición en zig-zag (a uno alto le sigue uno más bajo, a éste uno más alto, etc) para que cuando la cámara les enfoque individualmente al sonar el himno nacional tenga que hacer un ridículo movimiento de vaivén. Suponiendo que el dorsal de los jugadores esté dado en función de la altura (1 para el jugador más bajo, 2 para el siguiente más bajo, etc.), realizar un programa que escriba todas las posibles combinaciones válidas (en zig-zag) dado un determinado número (N) de jugadores. Por ejemplo, si N=4, las combinaciones válidas serían: 1324, 1423, 2143, 2314, 2413, 3142, 3241, 3412, 4132, 4231 Resolverlo mediante las estrategias:

- Generando recursivamente todas las permutaciones de N valores y filtrando aquellas que cumplan la condición de zig-zag. Nivel de dificultad: Difícil.
- Generando recursivamente todas las permutaciones en zig-zag de manera directa. Nivel de dificultad: Extremadamente difícil.

107.- **Laberinto (ACM 705)**. Una manera fácil de crear pequeños laberintos es rellenar un rectángulo usando los caracteres `'/'` y `'\'`. Por ejemplo:



Como se puede apreciar, los laberintos generados de esta forma no tienen bifurcaciones, por lo que pasando de una casilla a otra sólo se pueden seguir caminos cíclicos o salir del laberinto. Se pide crear un programa que pida el número de filas y columnas del laberinto (como máximo se puede considerar un laberinto de 20x20), y a continuación lea su descripción, la cual esta dada como una serie de líneas (tantas como filas tenga), cada una conteniendo caracteres `'/'` o `'\'`, tantos como columnas tenga. El objetivo del programa es contar el número de ciclos distintos que tiene y el número de casillas distintas que hay que recorrer en el ciclo más largo. Por ejemplo, el laberinto del gráfico correspondería a la entrada:

```
4 6
\ / / \ \ /
\ / / / \ /
/ / \ \ / \
\ / \ / / /
```

Y el resultado sería 2 ciclos distintos, el mayor de 16 casillas de longitud. Nivel de dificultad: Bastante difícil.

108.- Pantalla LCD. Se desea crear un programa que lea un número y lo escriba al estilo de las pantallas LCD, usando los caracteres '-' y '|'. Por ejemplo, si el número leído fuera el 14, la salida sería:

```
|    |_|
|      |
```

Nivel de dificultad: Asequible. Elementos de Programación: Arrays, Conjuntos

Ejercicios de Ficheros

109.- Escribir un programa que copie en un fichero del tipo file of tipo otros dos ficheros del mismo tipo, uno a continuación del otro.

110.- Elaborar un programa en Pascal que pida un valor entero al usuario y busque en un fichero de enteros dado dicho valor, y en caso de encontrarlo, que indique la posición en la que se ha encontrado.

111.- (Examen curso 87/88). Elaborar un programa en Pascal que calcule el porcentaje de números capicúas que existan en el fichero de tipo integer numeros.dat..

112.- Escribir un programa que lea valores de un fichero de arrays e imprima la media de cada array en un fichero de texto.

113.- Elaborar un programa que liste por pantalla un fichero de texto cualquiera

114.- Elaborar un programa Pascal que lea un fichero de texto en disco y cuente el número total de líneas y de caracteres del fichero.

115.- Dado un fichero de texto, escribir un programa Pascal que cuente el número de palabras que aparezcan en dicho fichero. Considerar como una palabra cualquier sucesión de caracteres separada por blancos o marca de fin de línea.

Ejemplo: (? significa fin de línea)

```
Este     es un ejemplo para  ?
Probar ?
El funcionamiento de un programa que cuenta las palabras
```

En este ejemplo la salida esperada es 15:

116.- (Examen curso 86/87). Un fichero almacena una sucesión de números enteros positivos o nulos, correspondientes a las puntuaciones de varios jueces sobre un mismo ejercicio.

- Escribir un programa para calcular el número de valores cero y la media aritmética de los valores estrictamente positivos, excepto el máximo y el mínimo de dichos valores positivos (suponiendo que sólo hay un máximo y un mínimo).
- Variar el programa para el caso en el que los valores máximo y mínimo puedan aparecer varias veces.

Nota: El fichero debe recorrerse una sola vez.

117.- Examen curso 87/88). De los distintos hospitales de la ciudad de Boston se han recogido datos a fin de estudiar el efecto que producen diversos tratamientos sobre ciertas enfermedades cancerígenas. Supuestas las siguientes definiciones de tipo:

```
type
   THospital = (HC, BosH, MassHC, Univ, USS);
   TEnfermedad = (leucem, piel, estom, pulm, larin);
   TTratamiento = 1..6;
   TBloque = array[THospital, TEnfermedad, TTratamiento] of integer;
```

Se han estructurado los datos en una matriz de tipo TBloque, de modo que en la componente de índices i, j, k se encuentra el número de enfermos curados en el hospital i, de la enfermedad j, mediante el tratamiento k. Esta matriz está grabada como primera componente del fichero estudio.dat de tipo file of TBloque. Elaborar un programa Pascal que presente en pantalla la siguiente información: :

- Número total de enfermos curados de la enfermedad j mediante el tratamiento k. Debe aparecer un listado que recoja todas las posibles combinaciones de enfermedad y tratamiento; cada una de sus líneas debe ser de la forma: <ENFERMEDAD ><TRATAMIENTO ><NUMERO DE CURADOS >de modo que se formen tres columnas bien alineadas encabezadas por un título.
- Número total de enfermos curados en cada uno de los hospitales. Debe darse una presentación similar a la pedida en el apartado anterior.

118.- (Examen curso 87/88). Dado un fichero de texto con nombre texto.dat, elaborar un programa Pascal que efectúe el recuento del número de líneas del fichero que tienen cero caracteres, un carácter, dos, tres, etc. y presente por pantalla un listado que recoja estos resultados. Sólo debe aparecer en el listado el número de líneas con j caracteres si hay al menos una línea con j caracteres. Por ejemplo, si el contenido del fichero fuera:

```
FECHA DEL EXAMEN : <eoln>
<eoln>
dia cuatro <eoln>
de febrero <eoln>
<eof>
```

el listado debería ser :

```
Caracteres :  0     Líneas :  1
Caracteres : 10     Líneas :  2
Caracteres : 17     Líneas :  1
```

Nota: Se considerará que la longitud máxima de una línea es de 255 caracteres.

119.- Examen curso 86/87). Dada una palabra, diremos que está escondida en un texto si está contenida en éste, no necesariamente de forma continua. Por ejemplo, la palabra "elefante" está escondida en la frase "El hermano de Francisco tuvo suerte". Elabórese un programa Pascal tal que, dada una palabra leída por teclado, la busque en un texto almacenado en el fichero texto.dat de tipo text. Como resultado, el programa debe exponer el mensaje:

```
Palabra <palabra> no encontrada en el texto
```

o bien

```
Palabra <palabra> encontrada escondida entre los <p> primeros
caracteres del texto
```

En el ejemplo, el resultado debería ser:

```
Palabra <elefante> encontrada escondida entre los 32 primeros
caracteres del texto
```

Téngase en cuenta la equivalencia entre minúsculas y mayúsculas. Resolver el problema de las siguientes formas:

- Suponiendo que la palabra a buscar no tiene más de 20 caracteres.
- Suponiendo que la palabra a buscar puede ser arbitrariamente grande. En este caso se puede suponer que la aparición de un asterisco (*) marca el final de la palabra.

Nota: El fichero debe recorrerse una sola vez.

120.- (Examen curso 88/89). Una determinada institución bancaria está interesada en conocer el número de créditos concedidos, de los diferentes tipos con los que ella trabaja, en cada una de las sucursales de las ciudades en que tiene abiertas oficinas, en un determinado intervalo de tiempo. Se supone que:

- El banco está implantado en un número variable de ciudades y en cada ciudad tiene abiertas cuatro sucursales: Urbana 0, Urbana 1, Urbana 2 y Urbana 3.
- Los tipos de crédito con los que trabaja son: Personal, Hipotecario, Preferencial y Otros tipos. Con esta información, realizar lo siguiente:
- Definir la estructura de fichero adecuada para almacenar dicha información.
- Elaborar un programa Pascal que cree y cargue datos en el fichero definido en el apartado anterior, suponiendo que los datos se capturan por teclado.

- Una vez creado el fichero anterior, elaborar un programa para calcular el número total de créditos concedidos de cada tipo y escribirlos, con un mensaje adecuado, en un fichero de texto creado al efecto.

121.- (Examen curso 88/89). Dado un fichero de números enteros, numeros.dat, almacenar en un fichero de texto, a razón de una posición por línea, las posiciones en que se encuentra el valor máximo.

122.- (Examen curso 90/91). Dado un fichero de matrices de orden 10x10 y componentes enteras, elaborar un único programa Pascal para resolver las siguientes cuestiones:

- Obtener un fichero de vectores de 10 componentes enteras que contenga las dos diagonales de cada matriz.
- Crear un fichero de texto para almacenar los valores de ambas diagonales a razón de dos diagonales por línea.
- Hallar el número de matrices del fichero cuyas diagonales son idénticas.

123.- (Examen curso 91/92). Un método común para poner dígitos de comprobación consiste en multiplicar cada dígito del número, incluyendo el dígito de seguridad, por un factor de peso y sumar los productos. El dígito de seguridad se elige de modo que el número formado sea divisible por un número apropiado, como puede ser el 9. Por ejemplo, al número 3746 se le debe añadir el dígito de seguridad 1, pues:

Número	3	7	4	6	1	–>dígito de seguridad
Factores de peso	x9	x5	x3	x7	x1	
Producto	27	35	12	42	1	= 117, divisible por 9

El fichero de enteros enteros.dat está formado por números positivos menores de 10.000. Elaborar un programa Pascal que añada a cada uno de estos números su dígito de seguridad correspondiente calculado con los factores de peso del ejemplo y los introduzca en un fichero de vectores de 5 dígitos.

124.- (Examen curso 91/92). Elaborar un programa que nos informe de la frecuencia de pares de letras adyacentes en un fichero de texto texto.txt. Por ejemplo si en el fichero aparece la frase “Esta frase” el programa debería contar una aparición del par st o as entre otros, pero no del par af (las letras están separadas por un espacio en blanco). Nota: Se puede suponer que en el texto sólo intervienen como letras los caracteres ASCII de la A a la Z, y que no hacemos distinción entre mayúsculas y minúsculas.

125.- (Prácticas curso 91/92). En el archivo de tipo texto glosario.dat se encuentran una serie de abreviaturas en orden alfabético seguidas por su descripción en la línea siguiente. Diseñar y elaborar un programa Pascal que dándole una abreviatura del fichero conteste con su descripción completa. Si la abreviatura pedida no apareciese en el fichero el programa indicará la abreviatura anterior y posterior. Nota: Las abreviaturas tienen una longitud máxima de 10 caracteres, pero desconocemos la longitud máxima de las descripciones.

126.- (Examen curso 86/87) Dada una cadena de caracteres A de tipo string[60] y un fichero de texto F, constrúyase un subprograma que exponga por pantalla los caracteres que están en el fichero pero no en la cadena.

127.- (Examen curso 86/87) Dado un fichero de texto F cuyas líneas no tienen más de 255 caracteres, escribir un procedimiento que lea cada línea de F, e imprima en otro fichero del mismo tipo, G, la línea obtenida por la siguiente regla: Primero se escribe el primer carácter, luego el último, después el penúltimo, a continuación el segundo, tercero, antepenúltimo, ante-antepenúltimo, etc. Ejemplo: de “turismo” escribe “tomursi”; de “cartas” escribe “csaart”.

128.- (Examen curso 86/87) Se define el tipo siguiente:

```
type TCadena = string[30];
     TAlumno = record
                 Apellido1, Apellido2, Nombre : TCadena;
                 NotaJunio, NotaSept: (NP,SU,AP,NO,SO,MH);
                 NotaPrac : integer;
     end;
```

En un archivo de tipo file of TAlumno disponemos de la lista de los alumnos matriculados en una asignatura. Se quiere obtener un fichero de texto en el que aparezca en cada línea el nombre de un alumno con el formato:

```
Apellidos,      Nombre                              Nota de junio
------------------------------------------------------------------
```

Elabórese un programa que realice la tarea.

Nota: Las columnas deberán estar correctamente tabuladas y en el apartado nota de junio se expresará la nota con todos sus caracteres.

129.- Un vendedor de coches lleva registro de todas sus ventas en forma de un fichero de texto, con el siguiente formato para cada línea:

- Caracteres 1-8: matrícula del coche;
- Caracteres 9-17: marca;
- Caracteres 18-20: fecha de venta.

Cada apartado está ajustado a la izquierda en su campo. La fecha de venta está codificada como los últimos dígitos del año seguidos por una letra para el mes ('A' para enero, 'B' para febrero, etc.). Por ejemplo:

```
3142BC RENAULT   78A
V1900AA SEAT      99H
S0007Z  MERCEDES  02L
B3456JH PEUGEOT   88A
```

Escribir un programa que lea estos datos y escriba los detalles de todos los Renault vendidos desde una fecha inicial hasta una fecha final, indicadas ambas como cadenas de tres caracteres.

Nota: Se puede suponer que no se ha vendido ningún automóvil antes del año 1950, y que este programa dejará de estar en servicio antes del 2050.

130.- Se considera el fichero de texto cuenta.dat, cuyas líneas están organizadas de la siguiente manera:

Número de cuenta	Formado por los caracteres posiciones 1-10
Titular de la cuenta	Formado por los caracteres posiciones 11-75
Tipo de operación	Formado por los caracteres posiciones 76-76
Importe de la operación	Formado por los caracteres posiciones 77- <eoln>

Se supone que el fichero está ordenado según los valores crecientes de número de cuenta y que pueden existir, para una misma cuenta, varias operaciones, o sea, varias líneas. Elaborar un subprograma que obtenga el número medio de operaciones por número de cuenta.

131.- Diremos que un número es de clase n si puede descomponerse como producto de n números primos. Por ejemplo, para números menores o iguales que 20, tenemos la siguiente distribución:

Clase	Números
1	1,2,3,5,7,11,13,17,19
2	4,6,9,10,14,18
3	8,12,18,20
4	16

Dado un fichero cuyos elementos son del tipo TVector = array [1..10] of 1..20, elaborar un programa que imprima en pantalla el número de elementos del fichero que tienen números de las cuatro clases posibles.