

1.1. Fíjese en la siguiente definición de tipos y sentencias. ¿Qué respuesta es correcta?

```
TYPE
  TNombre = string[20];
  TEdad = 18 .. 65;
  TSueldo = real;
  TEmpleado = RECORD
      nombre: TNombre;
      edad: TEdad;
      sueldo: real;
  END;
  TTrabajadores = array [1..100] OF TEmpleado;
  TDuenio = RECORD
      nombre: TNombre;
      edad: TEdad;
      trabajadores: TTrabajadores;
  END;
VAR
  a: TEmpleado;
  b,c: TDuenio;
BEGIN
  ...
  b.trabajadores[25].nombre:= a.nombre;
  c:= a;
... END;
```

- (a) Las dos instrucciones son correctas.
  - (b) Ninguna instrucción es correcta, son tipos incompatibles.
  - (c) La primera instrucción es correcta y la segunda es incorrecta.
  - (d) La primera instrucción es incorrecta y la segunda correcta.
- 1.2. Indicar cuál de las siguientes afirmaciones sobre ficheros es cierta:
- (a) Los ficheros pueden pasarse como parámetros a subprogramas pero únicamente por referencia.
  - (b) La función FILESIZE(var\_fichero) devuelve el tamaño en Kilobytes del fichero que recibe como parámetro.
  - (c) La función FILEPOS(var\_fichero) devuelve la posición física, esto es el nombre del directorio donde se encuentra almacenado el fichero que recibe como parámetro.
  - (d) Ninguna de las anteriores.
- 1.3. Al efectuar la operación RESET con un fichero binario...:
- (a) Se pueden realizar operaciones sólo de lectura
  - (b) Se pueden realizar operaciones sólo de escritura
  - (c) Se pueden realizar operaciones de lectura y escritura
  - (d) Si el fichero que se intenta abrir no existe da un error en tiempo de compilación
- 1.4. Si se quiere concatenar el contenido de un primer fichero de texto al final del contenido de un segundo fichero de texto:
- (a) Se abre el primer fichero con RESET y el segundo con REWRITE y se va copiando el contenido del primer fichero línea a línea.

- (b) Se abre el primer fichero con RESET y el segundo con APPEND y se va copiando el contenido del primer fichero línea a línea.
- (c) La respuesta a) y b) son correctas.
- (d) La respuesta a) y b) son incorrectas.

1.5. ¿Cuál de las siguientes afirmaciones sobre los arrays en Pascal es cierta?

- (a) El tamaño de un array puede variar en tiempo de ejecución.
- (b) Un array es una estructura de datos que contiene elementos de un mismo tipo.
- (c) El tipo base de un array necesariamente tiene que ser un tipo de datos simple.
- (d) Ninguna de las anteriores.

1.6. EOLN es una marca que:

- (a) delimita el fin de fichero en cualquier tipo de ficheros
- (b) delimita el fin de línea en cualquier tipo de ficheros
- (c) delimita el fin de línea en los ficheros binarios
- (d) delimita el fin de línea en los ficheros de texto

1.7. Dado el siguiente subprograma:

```
PROCEDURE UnProc (VAR p: cadena; n:integer);
BEGIN
  IF n = 1 THEN write(p[1])
  ELSE BEGIN
    write(p[n]);
    UnProc(p,n-1);
  END;
END;
```

¿Qué se muestra por pantalla cuando se realiza la llamada unProc('abcde',5)?

- (a) edcba
- (b) eabcd
- (c) eedcba
- (d) Se producen infinitas llamadas recursivas

1.8. Dado el siguiente código de un subprograma:

```
PROCEDURE OtroProc (n:integer);
BEGIN
  IF n = 1 THEN write(n)
  ELSE BEGIN
    write(n mod 2);
    OtroProc(n div 2);
  END;
END;
```

Qué salida se mostrará por pantalla cuando se hace la llamada OtroProc(16):

- (a) 10000
- (b) 00001
- (c) 01111
- (d) 11110

### Juego de la Vida:

El juego de la vida es un autómata celular diseñado por J.H. Conway en 1970. Se trata de un solitario cuyo desarrollo simula en cierto modo el surgimiento, evolución y desaparición de una sociedad de organismos vivos. El juego se realiza sobre un tablero cuadrado bastante grande dividido en celdas o células. Las células tienen dos estados posibles: vivo o muerto. Cada célula tiene ocho células vecinas que son las que están próximas a ella, incluso en las diagonales. El juego consiste en partir de un estado inicial del tablero de juego y observar cómo evoluciona en cada paso aplicando las siguientes reglas que están relacionadas con el estado de las células vecinas:

- Nacimiento: Una célula muerta con exactamente 3 células vecinas vivas, nace (en la iteración siguiente estará viva).
- Supervivencia: Una célula viva con 2 o 3 células vivas, sigue viva.
- Muerte: En cualquier otro caso, la célula muere o permanece muerta.

Es importante observar que todos los nacimientos y muertes ocurren simultáneamente. Juntos constituyen una sola generación (una iteración) en la historia completa de la evolución del juego de la vida.

Para garantizar que cada celda tenga exactamente 8 vecinas asumiremos que la fila N está situada sobre la fila 1, la fila 1 está situada debajo de la fila N, la columna N está situada a la izquierda de la columna 1, y la columna 1 está situada a la derecha de la columna N.

Cada partida se puede acabar por dos motivos: todas las células se encuentran muertas o se ha alcanzado un número máximo de iteraciones por partida.

8Y								6Y	7Y
			1X	2X	3X				
			4X	X	5X				
			6X	7X	8X				
3Y								1Y	2Y
5Y								4Y	Y

Se pide:

- Definir las constantes y los tipos necesarios para implementar el juego de la vida teniendo en cuenta los requisitos de los apartados siguientes. **[5 pts]**
- Realizar un subprograma que calcule el número de vecinos vivos de una determinada célula del tablero de juego. **[15 pts]**

- c) Realizar un subprograma que realice la actualización del estado del tablero de juego teniendo en cuenta las reglas de evolución descritas anteriormente (nacimiento, supervivencia y muerte). Este subprograma deberá utilizar el subprograma del apartado anterior. **[10 ptos]**
- d) Realizar un subprograma que reciba un array unidimensional de registros. Cada uno de los registros guardados en el array contiene información sobre el número de células iniciales de la partida, el estado inicial del tablero de juego y el número de iteraciones en el que se ha llegado a la condición de finalización de la partida. El subprograma se encargará de ordenar el array de registros de menor a mayor por el número de iteraciones que se realizaron antes de finalizar la partida utilizando el método de la burbuja. **[15 ptos]**
- e) Realizar un subprograma que indique si dos tableros son iguales. **[5 ptos]**
- f) Realizar un subprograma recursivo que busque un determinado registro en una lista enlazada y determine su presencia o ausencia. Los registros contendrán el número de células iniciales de la partida, el estado inicial del tablero de juego y el número de iteraciones en el que se ha llegado a la condición de finalización de la partida. Dado que el subprograma debe comparar dos tableros iniciales, éste debe llamar al subprograma anterior. **[15 ptos]**
- g) Realizar un subprograma que reciba como parámetros un fichero binario y un registro. El registro contiene información sobre el número de células iniciales de la partida, el estado inicial del tablero de juego y el número de iteraciones en el que se ha llegado a la condición de finalización de la partida. El subprograma se encargará de abrir el fichero binario 'resultados.dat' e insertar al final del mismo la información del registro sin destruir los datos existentes en el fichero. **[15 ptos]**

Ejemplo de la evolución de una partida del juego (células vivas \*, muertas por -) para un tablero de 5 filas y 5 columnas y 10 células inicialmente vivas

-	*	-	-	*
-	-	*	*	-
-	*	*	-	-
*	-	*	-	*
-	*	-	-	-

*	*	-	*	-
*	-	-	*	-
*	-	-	-	*
*	-	*	*	-
-	*	*	*	*

-	-	-	-	-
-	-	*	*	-
*	-	*	-	-
-	-	-	-	-
-	-	-	-	-

-	-	-	-	-
-	*	*	*	-
-	*	*	*	-
-	-	-	-	-
-	-	-	-	-

## **Biblioteca:**

Queremos realizar un programa de gestión de una biblioteca. La biblioteca guardará información sobre libros utilizando un array parcialmente lleno (formado por un array y un entero que indique el número de elementos que contiene el array). La información guardada será almacenada en registros (uno por cada libro de la biblioteca). La información que se guarda sobre cada uno de los libros de la biblioteca será: i) el ISBN (DNI de los libros - Es una cadena de 10 caracteres numéricos), ii) autor del libro, iii) título del libro y iv) número de páginas. **NOTA:** Se supone que sólo puede haber un **único ejemplar de cada libro** en la biblioteca.

Se pide:

- a) Definir los tipos de datos necesarios para realizar **TODOS** los apartados [5 pts].
- b) Subprograma que determine la posición de un libro con un determinado ISBN en la biblioteca. Si el libro no existe en la biblioteca se devolverá un 0 [10 pts].
- c) Subprograma que realice la inserción de un nuevo libro en la biblioteca (no necesariamente en orden). Se deberá comprobar que el libro no se encuentre ya insertado en la biblioteca (para ello se llamará al subprograma especificado en el apartado anterior, por lo que la comprobación se realizará por el ISBN del libro). Si ya se encuentra insertado o si la biblioteca está llena se devolverá un código de error [5 pts].
- d) Subprograma que realice la modificación de los datos de un libro. Desde el subprograma se pedirá al usuario que introduzca el ISBN del libro a modificar. Se podrán modificar todos los campos excepto el ISBN del libro. La modificación sólo se podrá realizar si el libro existe en la biblioteca [10 pts].
- e) Subprograma que ordene los libros de la biblioteca según el nombre de autor de menor a mayor por el método de **selección directa** [15 pts].
- f) Realizar un subprograma que se encargue de almacenar los libros de la biblioteca en un **fichero binario**. Los datos de los libros almacenados en el fichero deberán estar **ordenados por nombre de autor** [5 pts].
- g) Realizar un subprograma que abra un **fichero binario** que contiene datos de libros y los inserte en la biblioteca. Si se produce un error al realizar la inserción se deberá avisar de ello y continuar intentando cargar el resto de datos. [10 pts]
- h) Reserva de un libro de la biblioteca. El sistema de reservas de la biblioteca se realizará utilizando una cola implementada con una **lista enlazada** como la representada en la siguiente figura (**no es una lista doblemente enlazada, ni tampoco existe un puntero apuntando al final de la lista**).



Se pide realizar el subprograma **RECURSIVO** que reciba el ISBN del libro a reservar y el usuario que lo reserva, y lo **inserte por el final de la cola del sistema de reservas** de libros [20 pts].