

Nombre: DNI:

Convocatoria: **Febrero 1ºPP** Semana: **Reserva Unión Europea** Tipo de Examen: **C**

- El **test** debe ser contestado **en la hoja de lectura óptica**. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta.
- El test es eliminatorio y aporta un 30% de la nota final. Son necesarias 8 respuestas correctas (6 con las prácticas aprobadas) para que se corrija el ejercicio.
- Cada respuesta correcta: 1 punto. Respuesta incorrecta o en blanco: 0 puntos.

1.- La función predefinida que permite conocer la posición de un valor en la lista de valores es:

- VAL
- ODD
- HIGH
- ORD

2.- Dada la siguiente declaración:

```
PROCEDURE Simetrico(texto : ARRAY OF CHAR);
```

- El argumento está bien definido
- Al argumento le falta el rango del índice
- No se puede pasar un vector como argumento
- El argumento siempre debe pasarse por referencia

3.- En un vector de caracteres o ristra (string):

- El primer elemento es siempre el de índice igual a cero
- El primer elemento es el carácter nulo (0C)
- No se pueden almacenar ristras de diferentes longitudes
- No se pueden incluir comillas en su interior

4.- La búsqueda por dicotomía requiere:

- Una variable centinela del tipo del elemento buscado
- Tres variables del tipo índice auxiliares
- Una variable BOOLEAN para saber el final de la inserción
- Dos variables del tipo índice para el recorrido

5.- ¿Qué valor tiene x al final del siguiente código?:

```
VAR x : INTEGER;  
PROCEDURE Suma(VAR dato : INTEGER);  
BEGIN  
    x := 5;  
    dato := 10 DIV dato;  
END Suma;  
....  
x := 0;    Suma(x);
```

- x vale 5
- x vale 0
- x vale 2
- Error división por 0

6.- Sean los siguientes conjuntos:

X := Colores{Rojo, Amarillo, Verde}

Y := Colores{Amarillo, Azul};

La operación que devuelve TRUE es:

- X >= Y
- X - Y
- X <> Y
- X / Y

7.- En Modula-2, los conjuntos:

- Pueden tener elementos repetidos
- Sólo se definen sobre tipos enumerados
- Forman parte del lenguaje
- Disponen de relaciones de orden entre sus elementos

8.- El desarrollo de programas por refinamientos sucesivos consiste en:

- Programar sucesivas versiones que intentan resolverlo
- Programar la solución hasta conseguir la solución refinada
- Descomponer en acciones repetibles
- Descomponer en acciones simples

9.- El siguiente módulo:

```
DEFINITION MODULE miModulo;  
    TYPE Dato;  
    PROCEDURE Operar(miDato : Dato) : Dato;  
END miModulo;
```

- Es una abstracción funcional
- Es un tipo opaco
- Es un dato encapsulado
- Es incorrecto

10.- En un registro con variantes el tipo del discriminante siempre tiene que ser:

- Cardinal
- Ordinal
- Real
- Entero

EJERCICIO DE PROGRAMACIÓN

Realizar el tipo abstracto de datos (TAD) **PrecioGlobal** para guardar el precio de un producto con hasta 4 monedas diferentes. La operación **NuevoProducto** guarda el NombreProducto pasado como argumento y pone a cero los precios en las 4 monedas. La operación **NuevoPrecio** con los argumentos NombreProducto, TipoMoneda (enumerado: euro, dólar, libra, yen) y PrecioMoneda, guarda el nuevo precio del producto en la moneda. La operación **ListaPrecios** escribe en pantalla el producto y los precios en las distintas monedas para el NombreProducto pasado como argumento.