

RELLENE EN ESTA HOJA Y EN LA HOJA DE LECTURA ÓPTICA LOS SIGUIENTES DATOS:

Apellidos:.....Tlfno.:.....

Nombre:.....D.N.I.:.....

Convocatoria: **Septiembre 1ªPP** Semana: **1ª** Tipo de Examen: **B**

- 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.
- La solución del ejercicio se realizará en el reverso de esta hoja. **No se corregirán hojas auxiliares.**

ENTREGUE ÚNICAMENTE ESTA HOJA Y LA HOJA DE LECTURA ÓPTICA sin grapar

TEST (cada respuesta correcta: 1 punto; respuesta incorrecta o en blanco: 0 puntos)

1.- En la declaración:

TipoSudoku = ARRAY Largo, Ancho OF CHAR;

- Largo y Ancho son de tipos simples**
- Largo y Ancho son de cualquier tipo
- Largo y Ancho son del mismo tipo
- Largo y Ancho son de tipos compatibles

2.- Las sentencias:

A:= 'B'; Write(A); Write('A');

imprimen la cadena:

- BA**
- BB
- AB
- AA

3.- La asignación A:=B; produce un error si:

- Si B es de tipo CARDINAL y A es de tipo INTEGER**
- Si B es de tipo INTEGER y valor negativo y A es de tipo CARDINAL
- Si B es de tipo INTEGER y valor positivo y A es de tipo CARDINAL
- Si A y B son de tipo INTEGER y su valor negativo

4.- Dada la siguiente declaración en Modula-2:

TYPE UnTipo = (uno, dos, tres, cuatro, cinco);

OtroTipo = SET OF UnTipo;

- Es correcto utilizar: UnTipo{dos .. cinco}
- Es correcto utilizar: OtroTipo{cuatro..dos}**
- Es correcto utilizar: ORD(OtroTipo)
- Es correcto utilizar: ORD(UnTipo)

5.- Con la ejecución del fragmento de Modula-2:

FOR i := 1 TO 5 DO

FOR j := i-1 TO 1 BY -1 DO WriteInt(j,1)

- El resultado es: 213214321
- El resultado es: 1213214321**
- El resultado es: 21321432154321
- El resultado es: 12123123412345

6.- Una función produce efectos laterales:

- Cuando no tiene argumentos
- Cuando devuelve un entero
- Cuando modifica el valor de variables externas**
- Cuando no devuelve nada

7.- ¿Cuál de los siguientes NO es un identificador válido?.

- RETUR
- Begin
- BEGIN**
- Return

8.- El desarrollo por refinamientos sucesivos:

- Sólo es aplicable al modelo imperativo de cómputo.
- Impide la compilación separada.
- Facilita resolución de problemas complejos dividiéndolos en subproblemas
- Exige el uso de intérpretes

9.- Dadas las siguientes declaraciones:

TYPE MiTipo = RECORD x, y: BOOLEAN; END;

VAR var: MiTipo;

PROCEDURE Inverso(VAR x: MiTipo; y: BOOLEAN); BEGIN

x.x:=VAL(BOOLEAN, ORD(x.x)-1);

var.y:=VAL(BOOLEAN, ORD(y)+1);

END Inverso;

Tras ejecutar el código:

var.x := TRUE; var.y := FALSE;

Inverso(var, var.y);

el valor de:

- var.x es FALSE y var.y es FALSE
- var.x es TRUE y var.y es FALSE
- var.x es TRUE y var.y es TRUE
- var.x es FALSE y var.y es TRUE**

10.- En el fragmento de código:

DEFINITION MODULE Radical

TYPE TipoRadical;

El "TipoRadical" es un:

- Tipo Abierto
- Tipo Opaco**
- Dato Encapsulado
- Dato Opaco

EJERCICIO DE PROGRAMACIÓN

Realizar un tipo abstracto de datos para resolver un sistema de traducción automático entre dos idiomas. El tipo abstracto almacenara las palabras equivalentes en dos idiomas, por ejemplo, para el francés-inglés, almacenará pares como por ejemplo: (voiture,car) o (maison,house), etc. El tipo abstracto dispondrá de dos operaciones el procedimiento AlmacenarPar que permite introducir un par de palabras equivalentes y la función Traducir que tiene como argumentos un idioma y una palabra, y devuelve la palabra traducida en el otro idioma disponible. (NOTA: para realizar la comparación de dos palabras se debe usar la función Comparar del módulo CADENA que recibe dos palabras y devuelve un valor TRUE si son iguales y FALSE si son diferentes).

RECUERDE: La solución del ejercicio se realizará en el reverso de esta hoja. NO se corregirá lo que exceda de este espacio.