

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

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

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

Convocatoria: **Febrero 1ªPP** Semana: **2ª** Tipo de Examen: **F**

- 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.- El desarrollo de programas por refinamientos sucesivos:
  - A. Es conveniente para construir programas complejos
  - B. No es conveniente si se utiliza un intérprete
  - C. Es adecuado para construir programas eficientes en tiempo de ejecución
  - D. No es apropiado cuando las variables tienen un gran tamaño
- 2.- La transparencia referencial de una función:
  - A. Se asegura con el desarrollo por refinamientos sucesivos
  - B. Sólo se consigue si todos sus parámetros se pasan por referencia
  - C. No puede lograrse si la función usa variables globales
  - D. Es contraria al principio de ocultación
- 3.- Dado el siguiente código MODULA-2, que podríamos decir de la última asignación.

```
VAR nombre : ARRAY [0..40] OF CHAR;  
VAR alias: ARRAY [0..20] OF CHAR;  
...  
nombre := "paquito";  
alias := nombre;
```

  - A. Error, incompatibilidad de tipos.
  - B. Se puede hacer porque nombre contiene un string de longitud 7 y por tanto cabe en la variable alias.
  - C. No está definida la asignación entre strings.
  - D. El tamaño del string *alias* es más pequeño que el de *nombre* y daría error.
- 4.- Los tipos subrango se utilizan para definir
  - A. ni nuevos tipos ni nuevos valores
  - B. nuevos tipos pero no nuevos valores
  - C. nuevos tipos y nuevos valores
  - D. nuevos valores pero no nuevos tipos
- 5.-¿Cuántas veces se muestra el mensaje "Hi"?

```
FOR d := 3 TO 5 DO  
  FOR f := 2 TO 5 DO  
    WriteString ("Hi");WriteLn; END; END;
```

  - A. 6
  - B. 12
  - C. 10
  - D. 15
- 6.- En Modula-2, el fragmento de programa:

```
Texto : ARRAY OF TipoElemento;
```

  - A. Es la declaración de un argumento de un PROCEDURE
  - B. Es la declaración de un tipo
  - C. Es la declaración de una variable
  - D. Es incorrecto
- 7.- En el fragmento de programa de Modula-2:

```
Tabla { 1..5, 10..15, 20..25 };
```

  - A. Tabla es una variable SET OF CARDINAL
  - B. Tabla es un tipo SET OF CARDINAL
  - C. Tabla es una variable conjunto
  - D. Tabla es un tipo conjunto
- 8.- El fragmento de MODULA-2

```
Z:= Y;
```

  - A. Equivale a Y:= Z.
  - B. Es una expresión booleana.
  - C. Z y Y deben ser números.
  - D. Es una sentencia.
- 9.- Tras la ejecución del siguiente código:

```
PROCEDURE Intercambiar(VAR y: INTEGER;  
x:INTEGER);  
VAR z: INTEGER;  
BEGIN  
  z:=x; x:=y; y:=z;  
END Intercambiar;  
  
VAR x,y: INTEGER;  
BEGIN  
  x:=1; y:=2;  
  Intercambiar(x,y);
```

  - A. x vale 2, y vale 1
  - B. x vale 2, y vale 2
  - C. x vale 1, y vale 1
  - D. x vale 1, y vale 2
- 10.- En Modula-2, la sentencia:

```
TYPE TipoAlgo;
```

  - A. Es correcta siempre
  - B. Es correcta dentro de un módulo de implementación
  - C. Es correcta dentro de un módulo de definición
  - D. Es correcta dentro del módulo principal

**EJERCICIO DE PROGRAMACIÓN**

Realizar un Tipo Abstracto de Datos para el registro de hasta 1000 reclamaciones. La información a guardar para cada reclamación es: Número de reclamación, Nombre, apellidos y teléfono de la persona que reclama, fecha y hora de la presentación de la reclamación y texto de la reclamación (200 caracteres), fecha y hora de la resolución de la reclamación y texto de la resolución (200 caracteres). Las operaciones a implementar son: Nueva reclamación, Listado de los números de reclamaciones no resueltas, Resolución de reclamación. NOTA: Se dispone de un módulo Reloj con la operación FechaHora para obtener la fecha y hora del reloj del ordenador.

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