

SOLUCIONES AL FINAL DEL EJERCICIO

Carrera: **I.T.I. de Gestión/Sistemas (Plan viejo/nuevo)**
Asignatura: **Programación I**
Material auxiliar: Ninguno

Fecha: 26 de enero del 2004
Hora: 11:30
Duración: **2 horas**

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

Apellidos:..... Tfno.:.....

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

Código Carrera:	40 (Sistemas plan antiguo)	Código Asignatura: 103	Convocatoria: Febrero 1ª PP
	41 (Gestión plan antiguo)		Semana: 1ª
	53 (Sistemas plan nuevo)	Código Asignatura: 103	
	54 (Gestión plan nuevo)		Tipo de Examen: A

- 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 40% de la nota final. Son necesarias 7 respuestas correctas (5 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.- ¿Cuál de las siguientes afirmaciones es FALSA?

- A. todo dato encapsulado es de tipo abstracto
- B. todo tipo opaco es tipo abstracto
- C. todo tipo abstracto es dato encapsulado
- D. con los módulos se definen tipos abstractos

2.- La palabra clave FUNCTION en Modula-2:

- A. no existe
- B. se utiliza para declarar funciones
- C. se utiliza para declarar procedimientos y funciones
- D. se puede usar igual que PROCEDURE

¿Ha codificado en la hoja de lectura óptica el campo convocatoria Febrero-1ª semana?

3.- Señale la impresión que produce el siguiente código:

```
cont := 0;
WHILE cont < 2 DO
  FOR cont:=1 TO 3 DO
    WriteInt(cont, 2);
  END; (* FOR *)
  INC(cont);
END; (* WHILE *)
```

- A. 1 2 3 1 2 3
- B. 1 2 3
- C. 0 1 2 3
- D. 0 1 2 3 0 1 2 3

4.- Si a:=TRUE, b:=FALSE y c:=FALSE, señale la expresión verdadera:

- A. a AND NOT c OR b
- B. NOT(NOT c OR b AND a)
- C. a AND NOT(a OR NOT c)
- D. NOT (NOT a OR c OR NOT b)

5.- ¿Cuál de las siguientes expresiones es correcta?

- A. ((5*30) DIV 6) MOD -5
- B. 33 / 5,3
- C. 25 * 3,5
- D. 33,5 DIV 5

6.- Los argumentos de tipo vector abierto:

- A. sólo pueden pasarse por valor
- B. sólo pueden pasarse por referencia
- C. sólo pueden utilizarse en funciones
- D. pueden pasarse por valor y por referencia

7.- En el siguiente fragmento de código en Modula-2:

```
.....
Algo.Local;
Otro.Remoto;
```

- A. Local puede ser una variable
- B. Remoto es una función
- C. Otro es un módulo
- D. Algo puede ser un registro

8.- En la sentencia: NEW(p);

- “p” es:
- A. un tipo puntero a cualquier tipo
 - B. una variable de cualquier tipo
 - C. un tipo simple declarado previamente
 - D. una variable de tipo puntero a cualquier tipo

9.- ¿Es posible utilizar más de una sentencia RETURN en una misma función?

- A. Sí, la ejecución acaba cuando se ejecuta cualquiera de las sentencias RETURN
- B. No, habría sentencias que nunca se ejecutarían
- C. Sí, porque el RETURN se ejecuta siempre que quiero escribir algo
- D. No, sólo si son procedimientos y no funciones

Antes de contestar a esta pregunta, asegúrese de haber completado sus datos personales en esta hoja.

10.- Un tipo enumerado

- A. pertenece a la clase de tipos ordinales
- B. es un subconjunto de un tipo predefinido del lenguaje
- C. se define a partir de otro tipo ordinal ya definido
- D. es un tipo estructurado de datos

EJERCICIO DE PROGRAMACIÓN

Un alumno de la UNED está identificado por un año de matrícula, un código de carrera y su NIF. Cada NIF (número de identificación fiscal) tiene asignada una letra. La asignación se realiza a partir del resto de la división del DNI entre 23. A cada uno de los posibles valores obtenidos le corresponde una letra del alfabeto (al 0 le corresponde la T, al 1 la R, al 2 la W, etc.). Se proporciona el módulo “NIF” con la función “AsignarLetra”, que nos da la letra que corresponde a cada uno de los restos posibles del 0 al 23.

Se pide:

- Realizar un tipo abstracto de datos, TipoAlumno, con procedimientos o funciones para:
 - o PedirAlumno: introducir por teclado la información necesaria (año, código y DNI) que identifica a un alumno.
 - o EscribirAlumno: escribir en la pantalla **TODA** (año, código y NIF) la información de un alumno.
- Escribir la declaración de “AsignarLetra” del módulo NIF (sólo los alumnos del plan viejo).

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

EXAMENES RESUELTOS DE PROGRAMACION I

Comentarios por Jose Antonio Vaqué

Soluciones oficiales

1.- C *Texto base, apartado 14.3, páginas 408 y siguientes.*

2.- A. *Texto base, apartado 3.2. El vocabulario de Modula-2, páginas 53 y siguientes.*

3.- B. Para contestar, vemos lo que hace el programa, o realizamos la traza del mismo. En este caso, con verlo es suficiente.

- Primero pone cont a cero, y entra en un bucle del que saldrá cuando cont sea mayor de 2.
- Luego entra en un bucle FOR, que hace tomar a cont los valores 1,2 y 3. En su interior imprime esos valores. Al finalizar, cont vale mas de 3.
- Luego incrementa cont, por lo que si valía 4, ahora vale 5.
- Encuentra el END del WHILE, y ahora cont ya es mayor de 2, y se sale de la ejecución

4.- A. *Texto base, apartado 52. Expresiones condicionales, páginas 104 y siguientes.*

Para contestar solo hay que seguir las operaciones de una en una, usando la precedencia descrita en la página 106 del libro, que se resume en este orden PARENTESIS, NOT, AND y OR, y dentro del mismo orden de izquierda a derecha. Pongo T para TRUE y F para FALSE:

- a) T AND NOT F OR F
T AND T OR F
T OR F
T
- b) NOT(NOT F OR F AND T)
NOT(T OR F AND T)
NOT(T AND T)
NOT T
F
- c) T AND NOT(T OR NOT F)
T AND NOT(T OR T)
T AND NOT T
T AND F
F
- d) NOT (NOT T OR F OR NOT F)
NOT (F OR F OR T)
NOT (F OR T)
NOT T
F

5.- A. Si vemos los operandos, es muy sencillo ver que en b,c,d se mezclan enteros y reales, por lo que solo queda a, en la que son todos enteros.

6.- D. *Texto base, apartado 11.6 Argumentos de tipo vector abierto, páginas 310 y siguientes.*

7.- C. Aunque todas parecen ciertas, si fueran la a) o la d) debería usarse Algo.Local := cosa; Si fuera la b) sería justo lo contrario, cosa := Otro.Remoto;

8.- D. Intenta despistar al no definir previamente un tipo para p. No puede ser a) ya que se necesita una variable, no un tipo, b) no es correcta, debe ser un tipo puntero, c) no es buena ya que no es un tipo, sino una variable.

9.- A. Es uno de los usos corrientes, poder salir de una función o procedimiento en cualquier lugar de la misma.

10.- A.

SOBRE EL EJERCICIO

Lo único a resaltar es que como hay que calcular un resto, debemos usar un tipo INTEGER o CARDINAL, ya que no es posible usar MOD para números REAL. Pero si un DNI puede tener hasta 8 dígitos, no será posible usar un tipo INTEGER (rango -32.768 a +32.768) o CARDINAL (rango 0 a 65.536), ha que usar un LONGINTEGER (rango - 2.147.483.648 a + 2.147.483.648), aunque aun mas apropiado sería usar un LONGCARDINAL, que debe alcanzar con 32 bits un rango entre 0 y 4.294.967.296, mas que suficiente para un DNI, y no admite números negativos. Lo malo es que para imprimir valores de estos tipos, no podemos usar ReadInt o WriteCard, sino ReadLongInt, ReadLongCard, WriteLongInt, WriteLongCar, definidos igualmente en el módulo InOut.

No se si en el examen admitirán el uso de este tipo de variables. Hay otras soluciones usando REAL, que es simplemente usar la expresión siguiente para calcular el resto, dado un número REAL a, el resto de su división entre otro número REAL b es el producto de b por la parte fraccionaria de a. Así podemos usar la expresión $\text{Resto} := (a - [a]) * b$, en donde [a] significa la parte entera de a. El problema es calcular la parte entera de a, que se puede conseguir usando la función del sistema TRUNC(a), que retorna un número INTEGER, que volvemos a convertir a REAL usando $\text{Resto} := (a - \text{REAL}(\text{TRUNC}(a))) * b$; pero aquí el problema es que el valor de a no debe sobrepasar el valor de un número INTEGER, y eso no lo podemos asegurar, pero podemos ir restándole a la variable el valor REAL(MAX(INTEGER)), de forma que cuando en algún momento $a < \text{REAL}(\text{MAX}(\text{INTEGER}))$, ya podemos usar la fórmula sin problemas, aunque esto ralentizará mucho la ejecución, no nos piden optimizar, sino una solución válida al problema. Y todo esto porque a Modula le falta una cosa tan útil como calcular la parte fraccionaria de un número Real.