Cuadernillo de examen

ASIGNATURA:	**Fundamentos de Programación I**	CÓDIGO:	**106**
CONVOCATORIA:	**Febrero 2006**	PLAN DE ESTUDIOS:	**2000/2002**
CURSO:	**1º**	CURSO ACADÉMICO:	**2005/2006**
TURNO:	**Mañana**	PROGRAMA:	**Ingeniería Informática** **Ingeniería Técnica en Informática**
CARÁCTER:	**Cuatrimestral (Primer cuatrimestre)**	ESPECIALIDAD:	**Común**
DURACIÓN APROXIMADA:	**2 horas y media**		

Solución propuesta

Preguntas teórico-prácticas

1. Programación modular. Criterios para la descomposición de un programa en módulos. Mecanismos de intercambio de información entre el programa principal y los módulos: ámbito de variables y paso de parámetros.

Apuntes de clase y apartados 6.4 y 6.5 del libro de texto

Aplicación
Se desea realizar un módulo que devuelva la división entera de dos números enteros que se pasarán como argumentos y **sin utilizar el operador de división entera**. Codifique dos versiones del módulo, una utilizando un procedimiento y otra utilizando una función.

```
procedimiento DivisiónEntera(valor entero : a,b ; ref entero : cociente)
inicio
   cociente ← 0
   mientras a >= b hacer
      cociente ← cociente + 1
      a ← a –b
   fin_mientras
fin_procedimiento

entero función DivisiónEntera(valor entero : a,b)
var
   entero : cociente
inicio
   cociente ← 0
   mientras a >= b hacer
      cociente ← cociente + 1
      a ← a –b
   fin_mientras
   devolver(cociente)
fin_función
```

Puntuación: 1,5 punto

2. Enumere y describa el funcionamiento de los métodos de ordenación que conozca.

Apartado 10.2 del libro de texto

Aplicación
En un array desordenado de registros se almacenan los resultados de una prueba ciclista. Cada registro contiene información con el dorsal, el nombre del corredor y el tiempo en segundos obtenido en la prueba. Utilizando el **método de ordenación de Shell** obtenga la clasificación de la prueba (los datos ordenados de menor a mayor por el campo tiempo). Declare además las estructuras de datos utilizadas.

```
const
   MaxEl = ...
tipos
   registro = corredor
      entero : dorsal
```

```
      cadena : nombre
      real :  tiempo
    fin_registro

    array[1..MaxEl] de corredor = corredores

procedimiento OrdenaciónShell(ref corredores:v; valor entero : n)
var
   entero : i,j,incr
inicio
   incr ← n div 2
   mientras incr > 0 hacer
      desde i ← incr + 1 hasta n hacer
         j ← i - incr
         mientras j > 0 hacer
            si v[j].tiempo > v[j + incr].tiempo hacer
               intercambiar(v[j], v[j + incr])
               j ← j - incr
            si_no
               j ← 0
            fin_si
              fin_mientras
           fin_desde
           incr ← incr div 2
   fin_mientras
fin_procedimiento
```

Puntuación: 1,5 puntos

3. Archivos. Explique:

- La estructura lógica y la estructura física de un archivo.
- Tipos de organización de archivos.

Apartados 9.1, 9.2 y 9.4 del libro de texto

Aplicación
Se tienen los datos de la prueba ciclista de la aplicación del ejercicio anterior en un archivo secuencial. Codifique:

- Un procedimiento que cargue la información (dorsal, nombre y tiempo) en el array de registros.
- Un procedimiento que almacene el array ordenado del apartado anterior en un nuevo archivo secuencial.

```
procedimiento CargarCorredores(ref corredores : c; ref entero : n)
//La variable N se cargará con el número total de registros leídos
var
   archivo_s de corredor : A
   corredor : R //El registro que se leerá del archivo
inicio
   abrir(A,lectura,'CARRERA.DAT')
   leer(A,R)
   n ← 0
   mientras no fda(A) hacer
      n ← n + 1
      c[n] ← R
      leer(A,R)
   fin_mientras
   cerrar(A)
fin_procedimiento

procedimiento GrabarCorredores(valor corredores : c; valor entero : n)
var
   archivo_s de producto : A
   entero : i
```

```
inicio
   abrir(A,escritura,'CARRERA.DAT')
   desde i ← 1 hasta n hacer
      escribir(A,c[i])
   fin_desde
   cerrar(A)
fin_procedimiento
```

Puntuación: 1,5 puntos

Pregunta práctica

Una librería almacena en un array la información de los N libros que ha vendido en el mes de enero. Por cada libro almacena:

- ISBN (una cadena).
- Título (una cadena).
- Precio de cada ejemplar (un dato real).
- El número de ejemplares vendidos en cada uno de los días del mes (31 datos enteros).

Se pide:

a) Declarar las estructuras de datos necesarias para realizar cada uno de los siguientes apartados

```
const
   NumLibros = ...
tipos
   registro = libro
      cadena : ISBN, título
      real : precio
      array[1..31] de entero : ventas
   fin_registro

   array[0..NumLibros] de libro = libros

   //Para meter el número de ejemplares vendidos (apartado b) y
   //utilizarlo para sacar el "Top 10"
   array[1..NumLibros] de entero : vector
```

Puntuación: 0,5 puntos

b) Realizar un módulo que calcule el número de ejemplares vendidos de cada título a lo largo del mes.

```
procedimiento VentasPorLibro(valor libros : l; valor entero : n;
                             ref vector : total)
var
   entero : i,j
inicio
   desde i ← 1 hasta n hacer
      total[0] ← 0
      desde j ← 1 hasta 31 hacer
         total[0] ← total[0] + l[i].ventas[j]
      fin_desde
   fin_desde
fin_procedimiento
```

Puntuación: 1 punto.

c) Realizar un módulo que calcule la facturación de cada libro en cada uno de los días del mes.

```
procedimiento FacturaciónPorDía(valor libros : l; valor entero : n)
var
   entero : i,j
```

```
   real : facturaciónDía
inicio
   desde i ← 1 hasta 31 hacer
      facturaciónDía ← 0
      desde j ← 1 hasta n hacer
         facturaciónDía ← facturaciónDía + l[j].precio * l[i].ventas[j]
      fin_desde
      escribir(facturaciónDía)
   fin_desde
fin_procedimiento
```

Puntuación: 1 punto.

d) Realizar un módulo que devuelva qué libro ha realizado una facturación más alta y en que día se realizó.

```
procedimiento VentasMáximas(valor libros : l; valor entero : n;
                            ref entero : TitMax, DiaMax)
var
   entero : i,j
inicio
   TitMax ← 1
   DiaMax ← 1
   desde i ← 1 hasta n hacer
      desde j ← 1 hasta 31 hacer
         si l[i].precio * l[i].ventas[j] >
                l[TitMax] * l[TitMax].ventas[DiaMax] entonces
            TitMax ← i
            DiaMax ← j
         fin_si
      fin_desde
   fin_desde
fin_procedimiento
```

Puntuación: 1,5 puntos.

e) Realizar un módulo que genere la lista de los "Top 10" (10 libros más vendidos del mes).

```
procedimiento TopTen(valor libros:l; valor vector : v; valor entero : n)
//Vector es el array con las ventas mensuales de cada libro
var
   entero : i,j,min
   libro : auxL  //Para intercambiar los libros
   entero : auxV //Para intercambiar las ventas
inicio
   desde i ← 1 hasta n-1 hacer
      min ← i
      desde j ← i hasta n hacer
         si v[j] < v[min] entonces
           min ← j
         fin_si
      fin_desde
      //Se intercambian los elementos del array de ventas
      auxV ← v[i]
      v[i] ← v[min]
      v[min] ← auxV
      //También es necesario intercambiar el array de libros
      //para considerar dos arrays paralelos
      auxL ← l[i]
      l[i] ← l[min]
      l[min] ← auxL
   fin_desde
   //Los diez primeros elementos de la lista serán los libros más vendidos
```

```
    desde i ← 1 hasta 10 hacer
       escribir(l[i].título
    fin_desde
fin_procedimiento
```

Puntuación: 1,5 puntos.