Cuadernillo de examen

ASIGNATURA:	**Fundamentos de Programación I**	CÓDIGO:	**106**
CONVOCATORIA:	**Septiembre 2005**	PLAN DE ESTUDIOS:	**2000/2002**
CURSO:	**1º**	CURSO ACADÉMICO:	**2004-2005**
TURNO:	**Mañana**	PROGRAMA:	**Ingeniería Técnica en Informática** **Ingeniería Informática**
CARÁCTER:	**Cuatrimestral (1er. cuatrimestre)**	ESPECIALIDAD:	**Común**
DURACIÓN APROXIMADA:	**2,5 horas**		

Soluciones propuestas al examen

Preguntas teórico-prácticas

1. Programación estructurada. Teorema de la programación estructurada. Describa el funcionamiento de los distintos tipos de estructuras selectivas y represéntelas utilizando diagramas de flujo, diagramas N-S y pseudocódigo.

Apartados 2.3, 4.3, 4.4 y 4.5 del libro de texto

Aplicación

Escriba un fragmento de código que permita saber la población a partir de una variable (`CP`) que contiene un código postal. El algoritmo deberá indicar si el código postal corresponde a Madrid (280xx), Alcobendas (281xx), El Escorial (28280) o no corresponde a ninguna de las tres. Realice **dos versiones** del algoritmo, una utilizando una estructura selectivas doble y otra utilizando una estructura selectiva múltiple.

a) Con estructura selectiva doble

```
entero : cp
cadena : población
...
si cp > 28200 entonces
   población ← 'EL ESCORIAL'
si_no
   si cp > 28100 entonces
      población ← 'ALCOBENDAS'
   si_no
      población ← 'MADRID'
   fin_si
fin_si
```

b) Con estructura selectiva múltiple

```
//cp div 100 devuelve los tres primeros dígitos del código postal
según_sea cp div 100 hacer
   280 : población ← 'MADRID'
   281 : población ← 'ALCOBENDAS'
   282 : 'EL ESCORIAL'
fin_según
```

Puntuación: 1,5 puntos

2. Búsqueda interna. Enumere y explique el funcionamiento de los distintos métodos de búsqueda en arrays que conozca.

Apartado 10.3 del libro de texto

Aplicación

Se tiene un array desordenado con registros que contienen el expediente y la nota de una serie de alumnos. Diseñe una función que permita buscar un alumno a partir de su expediente utilizando el método de búsqueda secuencial con centinela. El array y el expediente se pasarán como argumentos en la función.

```
tipos
   registro = alumno
      cadena = expediente
      real : media
```

```
   fin_registro
   array[0..n] de alumno = alumnos
...
entero: función Buscar(E alumnos:v; E cadena:exp; E entero:n)
var
   entero: i
inicio
   i ← n
   v[0].expediente ← el
   mientras exp <> v[i].expediente hacer
      i ← i - 1
   fin_mientras
   devolver(i)
fin_función
```

Puntuación: 1,5 puntos

3. Concepto de archivo. Describa los distintos tipos de acceso y de organización de archivos. Describa las instrucciones para manejar archivos secuenciales.

Apartados 9.1, 92, 9.3 y 9.4 del libro de texto

Aplicación

Se tiene almacenado en un archivo secuencial información sobre una serie de personas con su nombre y la ciudad de residencia. Diseñe un algoritmo que almacene en otro archivo secuencial a todas las personas de Madrid.

```
algoritmo Ejercicio3
tipos
   registro = persona
      cadena : nombre, ciudad
   fin_registro
var
   archivo_s de persona : A1, A2
   persona : R
inicio
   abrir(A1, 'PERSONAS.DAT', lectura)
   crear('MADRID.DAT')
   abrir(A2, 'MADRID.DAT', escritura)
   leer(A1,R)
   mientras no fda(A1) hacer
      si R.ciudad = 'MADRID' entonces
         escribir(A2,R)
      fin_si
      leer(A1,R)
   fin_mientras
   cerrar(A1,A2)
fin
```

Puntuación: 1,5 puntos

Preguntas prácticas

Se tiene almacenado en un archivo secuencial los expedientes y las notas de las 7 asignaturas del primer cuatrimestre de N alumnos del primer curso. Se pide:

1. Declare las estructuras de datos necesarias para realizar los puntos siguientes.

```
const
   MaxAlum = ...  //Número máximo de alumnos
tipos
   registro = alumno
      cadena : exp
      array[1..7] de real : notas //Cada alumno tiene 7 notas
   fin_registro
```

```
   array[1..MaxAlum] de real = tabla

   registro = media
      cadena : exp
      real : media
   fin_registro
   array[0..MaxAlum] de media = vector
```

Puntuación: 0,5 punto.

2. Diseñe un módulo que permita almacenar las notas de todos los alumnos en una tabla de Nx7 elementos. Al mismo tiempo se deberá almacenar en un array de registros con los campos expediente y nota media, los expedientes de los alumnos y su nota media.

```
procedimiento CargarNotas(E/S tabla : t; E/S vector : v; E/S entero : n)
//el parámetro n devolverá en número de alumnos cargados
var
   archivo_s de alumno : A
   alumno : R
   entero : i  //Para recorrer las notas de cada alumno
inicio
   n ← 0   //Al arrancar se han cargado 0 alumnos
   abrir(A1, 'NOTAS.DAT', lectura)
   leer(A,R)
   mientras no fda(A) hacer
      n ← n + 1
      //Se inicializa el elemento n del array de registros de medias
      v[n].exp ← R.exp
      v[n].media ← 0   //La media es un acumulador y se inicializa a 0
      //Cargar la tabla y calcular la media
      desde i ← 1 hasta 7 hacer
         t[n,i] ← R.notas[n,i]
         v[n].media ← v[n].media + t[n,i]
      fin_desde
      media ← media / 7
      leer(A,R)
   fin_mientras
   cerrar(A)
fin_procedimiento
```

Puntuación: 2 puntos

3. Diseñe un módulo que permita ordenar el array de registros por el campo expediente.

```
procedimiento OrdenaciónInserciónDirecta(E/S vector:v;E entero : n)
//El vector v tiene elementos entre 0 y n
var
   entero : i,j
inicio
   desde i ← 2 hasta n hacer
      //Se almacena el elemento a insertar (v[i]) en la posición 0
      //del array para que actúe como centinela
      v[0] ← v[i]
      j ← i – 1
      //Se desplazan todos los elementos mayores que v[0]
      //y situados a su izquierda una posición a la derecha
      mientras v[i].exp > v[0].exp hacer
          v[j+1] ← v[j]
          j ← j – 1
      fin_mientras
      //En la posición siguiente al primer elemento menor o igual
      //se inserta el elemento v[0]
```

```
      v[j+1] ← v[0]
   fin_desde
fin_procedimiento
```

Puntuación: 1,5 puntos

4. Diseñe un módulo que permita obtener la nota media de cada una de las 7 asignaturas.

```
procedimiento MediaPorAsignatura(E tabla : t; E entero n)
var
   entero : i,j
   real : media
inicio
   desde i ← 1 hasta 7 hacer
      media ← 0
      desde j ← 1 hasta n hacer
         media ← media + t[j,i]
      fin_desde
      media ← media / n
      escribir('Media asignatura ',i, media)
   fin_desde
fin_procedimiento
```

Puntuación: 1,5 puntos