UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID
Facultad de Informática
Escuela Universitaria de Informática
Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

ASIGNATURA:	**Fundamentos de Programación I**	CÓDIGO:	**106**
CONVOCATORIA:	**Septiembre 2004**	PLAN DE ESTUDIOS:	**2000 / 2002**
ESPECIALIDAD:		CURSO:	**1º**
TURNO:	**Mañana**	CURSO ACADÉMICO:	**2003/2004**
CARÁCTER:	**Cuatrimestral (1er. cuatrimestre)**	PROGRAMA:	**Ingeniería en Informática / Ingeniería Técnica en Informática**

DURACIÓN APROXIMADA: **2 horas y media**

Soluciones propuestas

Preguntas teórico-prácticas

1. Estructuras de control repetitivas. Describa las estructuras de control repetitivas que conozca. Escriba su representación en diagrama de flujo y pseudocódigo. Indique en que ocasiones es recomendable utilizar cada una de las estructuras.

Capítulo 5 del libro de texto

Aplicación

Se desea escribir un programa que calcule la media de 100 números enteros leídos por teclado. Realice distintas versiones del programa utilizando las estructuras de control repetitivas que conozca.

```
algoritmo Ejercicio1A
//Utilizando una estructura desde
var
   entero : n, i
   real : media, suma
inicio
   suma ← 0
   desde i ← 1 hasta 100 hacer
      leer(n)
      suma ← suma + n
   fin_desde
   media ← suma / 100
   escribir(media)
fin

algoritmo Ejercicio1B
//Utilizando una estructura mientras
var
   entero : n, i
   real : media, suma
inicio
   suma ← 0
   i← 0
   mientras i < 100 hacer
      leer(n)
      i ← i + 1
      suma ← suma + n
   fin_mientras
   media ← suma / 100
   escribir(media)
fin

algoritmo Ejercicio1C
//Utilizando una estructura repetir
var
   entero : n, i
   real : media, suma
inicio
   suma ← 0
   i← 0
   repetir
      leer(n)
```

```
      i ← i + 1
      suma ← suma + n
   hasta_que i = 100
   media ← suma / 100
   escribir(media)
fin
```

Puntuación: 1,5 puntos

2. Cadenas de caracteres. Describa los distintos tipos de cadenas de caracteres según su forma de almacenarse en memoria.

Apartado 8.3 del libro de texto y apuntes de clase

Aplicación

Codifique una función que cuente el número de veces que aparece una cadena dentro de una cadena principal. Tanto la cadena buscada como la cadena principal se pasarán como argumentos de la función.

```
entero función ContarApariciones(E cadena : cp, sc)
//cp es la cadena principal, sc es la subcadena
var
   entero : conta,p
inicio
   conta ← 0
   p ← posición(cp,sc)
   mientras p <> 0 hacer
      conta ← conta + 1
      cp ← subcadena(cp,p+1)
      p ← posición(cp,sc)
   fin_mientras
   devolver(conta)
fin_función
```

Puntuación: 1,5 puntos

3. Archivos. Describa los distintos tipos de organización de archivos.

Apartado 9.4 del libro de texto

Aplicación

Suponiendo que está creado un archivo secuencial que posee, entre otros campos un campo de tipo cadena llamado `Código`, codifique un procedimiento que permita buscar un registro en dicho archivo. La búsqueda se hará a partir de un valor del código que se pasará como argumento al procedimiento. El procedimiento devolverá el resultado de la búsqueda mediante un valor lógico.

```
procedimiento Buscar(E cadena:código ; S lógico : encontrado)
var
   tipoRegistro : R     //Tipo de registro base del archivo
   tArchivo : A         //Tipo de archivo ya declarado
inicio
   abrir(A,"miArchivo.dat",lectura)
   leer(A,R)
   mientras (R.código <> código) y no fda(A) hacer
      leer(A, R)
   fin_mientras
   encontrado ← R.código = código
   cerrar(A)
fin_procedimiento
```

Puntuación: 1,5 puntos

Preguntas prácticas

Un array de N registros almacena los datos de una serie de productos. El array ya está cargado con los datos y por cada producto se guarda su Código, su descripción y su stock. De forma paralela se almacenan en una tabla de Nx7 elementos las cantidades vendidas de cada producto a lo largo de la semana, de forma que en el elemento i,j de la tabla aparecerán las unidades vendidas del producto i en el día j. Por ejemplo, del Bolígrafo Bic Azul (posición 2 del array de registros) existen en el almacén 325 unidades, y el miércoles (fila 2, columna 3 de la tabla) se vendieron 34 unidades.

PRODUCTOS

	Código	Descripción	Stock
1	…	…	…
2	30134	Bolígrafo Bic Azul	325
3	…	…	…
4	…	…	…
…	…	…	…
N	…	…	…

VENTAS

	1	2	3	4	5	6	7
1	…	…	…	…	…	…	…
2	10	12	34	31	54	10	1
3	…	…	…	…	…	…	…
4	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
N	…	…	…	…	…	…	…

1. Escriba la declaración de todas las estructuras de datos necesarias para realizar las acciones de los puntos siguientes.

Puntuación: 0,5 puntos

```
const
   N = …   //Número de productos a manejar
tipos
   registro = Producto
      cadena : código, descripción
      entero : stock
   fin_registro
   array[0..N] de Producto = Productos

   array[1..N,1..7] de entero = Ventas
```

2. Codifique un módulo que permita leer las ventas de cada producto, teniendo en cuenta que la entrada de datos se hará por día de la semana (es decir, primero se leerán las ventas del lunes del producto 1, después las ventas del lunes del producto 2, después las ventas del lunes del producto 3, etc.)

Puntuación: 1 punto

```
procedimiento LeerVentas(E/S ventas : tabla; E entero : n)
var
   entero : i,j
inicio
   desde i ← 1 hasta 7 hacer
      //Leer todas las ventas del día i
      desde j ← 1 hasta n hacer
         leer(tabla[j,i])
      fin_desde
   fin_desde
fin_procedimiento
```

3. Codifique un módulo que actualice el stock de cada producto, restando al stock actual las ventas realizadas a lo largo de la semana.

Puntuación: 1 punto

```
procedimiento ActualizarVentas(E ventas : tabla; E/S Productos : p)
var
   entero : i,j
inicio
   desde i ← 1 hasta n hacer
      desde j ← 1 hasta 7 hacer
         p[i].stock ← p[i].stock – tabla[i,j]
      fin_desde
   fin_desde
fin_procedimiento
```

4. Codifique un módulo que permita saber cual ha sido el producto más vendido a lo largo de la semana.

Puntuación: 2 puntos

```
cadena función ProductoMásVendido(E ventas : tabla ; E Productos : p)
//Devuelve la descripción del producto más vendido
var
   entero : i,j,suma,Max,PMax //Max guarda el total de ventas del producto más vendido
                              //PMax guarda la posición del producto más vendido
inicio
   Max = 0    //Se supone que al menos se ha vendido 1 unidad de algún producto
   desde i ← 1 hasta n hacer
      //Se suma el total de ventas del producto i
      suma ← 0
      desde j ← 1 hasta 7 hacer
         suma ← suma + tabla[i,j]
      fin_desde
      si suma > Max entonces
          Max ← suma
          PMax ← i
      fin_si
   fin_desde
   devolver(p[PMax].descripción)
fin_función
```

5. Codifique un módulo que ordene el array de registros de forma ascendente por el campo stock.

```
procedimiento Ordenar(E/S productos : p ;E entero : n)
//Ordenación por inserción directa
var
   entero : i,j
inicio
   desde i ← 2 hasta n hacer
      p[0] ← p[i]
      j ← i - 1
      mientras p[j].stock > p[0].stock hacer
          p[j+1] ← p[j]
          j ← j - 1
      fin_mientras
      p[j] ← p[0]
   fin_desde
fin_procedimiento
```

Puntuación: 1 punto