



ASIGNATURA:	Fundamentos de Programación I	CÓDIGO:	106
CONVOCATORIA:	Febrero de 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 horas y media		

Solución propuesta

Preguntas teórico-prácticas

1. Tipos de datos. Describa los distintos tipos de datos, utilizados en el lenguaje algorítmico y su utilidad, tanto simples como estructurados, estándar o definidos por el usuario.

Apartados 3.5, 7.1 del libro de texto y apuntes de clase

Aplicación

Se desea almacenar en una única estructura de datos la información sobre los corredores participantes en una carrera por etapas. En la carrera participan 120 corredores y por cada corredor se desea almacenar su dorsal, su nombre, el equipo al que pertenece, la fecha de nacimiento (años, mes y día) y los segundos que ha tardado en realizar cada una de las 25 etapas de la carrera. Realice las declaraciones de datos necesarias para definir la estructura de datos que almacene dicha información.

tipos

```
registro = fecha
    entero = dia, mes, año
fin_registro
registro = corredor
    entero : dorsal
    cadena : nombre, equipo
    fecha : fechaNacimiento
    array[1..25] de entero : tiempos
fin_registro
array[1..120] de corredor : carrera
```

Puntuación: 1,5 puntos

2. Cadenas de caracteres. Tipos de cadenas según la forma de almacenamiento en memoria. Operaciones con cadenas. Explique de forma detallada como se realizan las comparaciones de cadenas y los problemas que puede suponer. Funciones básicas para el tratamiento de cadenas.

Apartados 8.3, 8.4.2, 8.4.3 y 8.5 del libro de texto



Aplicación

Se tiene almacenada una frase en una variable de tipo cadena. Suponiendo que cada palabra está separada por un único espacio en blanco, con excepción de la última, codifique una función que reciba como argumento la cadena y devuelva el número de palabras que contiene.

```
entero función ContarPalabras(E cadena : c)
var
  entero : conta
inicio
  conta ← 0
  //Mientras exista un espacio en blanco
  mientras posición(c, ' ') <> 0
    conta ← conta + 1
    //Se extraen todos los caracteres a partir del espacio en blanco
    c ← subcadena(posición(c, ' ')+1)
  fin_mientras
  //Se cuenta la última palabra, si es que la cadena tiene algún carácter
  if c <> '' entonces //Si c es una cadena nula, la cadena estaba vacía
    //y conta debería valer 0
    conta ← conta + 1
  fin_si
  devolver(conta)
fin_función
```

Puntuación: 1,5 puntos

3. Ordenación de arrays. Enumere y describa brevemente los distintos métodos de ordenación de arrays que conozca.

Apartado 10.2 del libro de texto

Aplicación

Diseñe un procedimiento que permita ordenar los elementos de un array de números enteros comprendidos entre dos elementos del mismo. La posición de esos dos elementos se pasarán como argumentos al procedimiento.

```
//Ordenación por el método de selección directa
procedimiento Ordenar(E/S vector : v; E entero : ini, fin)
var
  entero : i, j, PMin, aux
inicio
  desde i ← ini hasta fin-1 hacer
    PMin ← i
    desde j ← i+1 hasta fin hacer
      si v[j] < v[PMin] entonces
        PMin ← j
      fin_si
    fin_desde
    aux ← v[i]
    v[i] ← v[PMin]
    v[PMin] ← aux
  fin_desde
fin_procedimiento
```

Puntuación: 1,5 puntos

Preguntas prácticas

Un concesionario de coches mantiene un archivo secuencial que contiene tanto los vehículos que se han vendido como los que le quedan en stock. Por cada vehículo se guarda la siguiente información:

- Identificador del vehículo.
- Marca y modelo
- DNI del comprador
- Año de la venta.



- Mes de la venta.

Nota: los vehículos que todavía no hayan sido vendidos tendrán vacíos los campos del DNI del comprador, y del año y mes de la venta.

Se pide:

1. Declarar todas las estructuras de datos necesarias para poder realizar **todas** las operaciones que se piden.

```
const
  MaxEl = ... //Número máximo previsto de registro en el archivo
tipos
  registro = coche
  cadena : id, marca, DNI
  entero : año, mes
fin_registro
array[0..MaxEl] de coche = coches
//Array para almacenar los totales del año 2004
array[1..12] de entero : ventas
var
  coches : c
  ventas : v
  entero : n //Número de registros real del archivo
```

Puntuación: 0,5 puntos.

2. Codificar un método que permita cargar en un array de registros todos los elementos del archivo secuencial.

```
procedimiento CargarDatos(E/S coches : c; E/S entero : n)
var
  archivo_s de coche : A
  coche : R
inicio
  abrir(A, lectura, 'COCHES.DAT')
  leer(A, R)
  n ← 0
  mientras no fda(A) hacer
    n ← n + 1
    c[n] ← R
    leer(A, R)
  fin_mientras
  cerra(A)
fin_procedimiento
```

Puntuación: 0,5 puntos.

3. Codificar un método que permita averiguar los coches que se han vendido cada mes a lo largo del año 2004.

```
procedimiento Ventas(E coches : c; E/S ventas: v; E entero : n)
var
  entero : i
inicio
  //Inicializar el array de ventas (todos sus elementos son contadores)
  desde i ← 1 hasta 12 hacer
    v[i] ← 0
  fin_desde

  desde i ← 1 hasta n hacer
    si c[i].año = 2004 entonces
      //Se incrementa el número de unidades del mes c[i].mes
      v[c[i].mes] ← v[c[i].mes] + 1
    fin_si
  fin_desde
fin_procedimiento
```

Puntuación: 2 puntos.



4. Codificar un procedimiento que ordene del array de forma descendente por el campo del DNI del comprador.

```
procedimiento Ordenar(E/S coches:c;E entero : n)
//El vector c tiene elementos entre 0 y n
var
    entero : i,j
inicio
    desde i ← 2 hasta n hacer
        c[0] ← c[i]
        j ← i - 1
        mientras c[i].DNI < c[0].DNI hacer
            c[j+1] ← c[j]
            j ← j - 1
        fin_mientras
        c[j] ← c[0]
    fin_desde
fin_procedimiento
```

Puntuación: 1 punto.

5. Codificar un procedimiento que permita guardar en un archivo secuencial los DNI de aquellos compradores que hubieran comprado más de un coche.

```
procedimiento GuardarDNIREpetidos(E coches : c; E entero : n)
var
    entero : i,conta
    archico_s de coche : A
    coche : aux
inicio
    crear('DNIREpetidos.DAT')
    abrir(A, escritura,'DNIREpetidos.DAT')
    //El array está ordenado descendentemente por DNI
    //Los coches sin ventas (con DNI vacío) están al final
    i ← 1
    mientras c[i].DNI <> '' hacer //Mientras sea distinto que una cadena nula
        conta ← 0
        coche ← c[i]
        mientras coche.DNI = (c[i].DNI) y (c[i].DNI <> '') hacer
            conta ← conta + 1
            i ← i + 1
        fin_mientras
        si conta > 1 entonces
            escribir(A,coche)
        fin_si
    fin_mientras
    cerrar(A)
fin_procedimiento
```

Puntuación: 1,5 puntos.