## *Cuadernillo de examen*

| | | | |
|---|---|---|---|
| ASIGNATURA | **Fundamentos de Programación I** | CÓDIGO | **106** |
| CONVOCATORIA | **Parcial de Diciembre de 2003** | PLAN DE ESTUDIOS | **2000** |
| ESPECIALIDAD | **Común** | CURSO | **1º** |
| TURNO | **Mañana** | CURSO ACADÉMICO | **2003/2004** |
| CARÁCTER | **Cuatrimestral** | PROGRAMA | **Ingeniería Técnica en Informática** |
| DURACIÓN APROXIMADA | **2 horas** | | |

# *Soluciones propuestas al examen*

## Preguntas teórico-prácticas

Estructuras de control repetitivas. Explique brevemente cada uno de los tipos de estructuras de control repetitivas, su formato del pseudocódigo y la utilidad de cada una de ellas.

*Capítulo 4 del libro de texto y apuntes de clase*

### Aplicación

Se desea realizar un algoritmo que permita sacar por pantalla los números pares entre 0 y n (n es un entero positivo introducido por teclado). Realice distintas versiones del algoritmo utilizando cada una de las estructuras de control repetitivas que conozca.

```
//Versión utilizando estructura desde
algoritmo NúmeroPares1
var
   entero : i, n
inicio
   leer(n)
   desde i ← 2 hasta n incremento 2 hacer
      escribir(i)
   fin_desde
fin

//Versión utilizando estructura mientras
algoritmo NúmeroPares2
var
   entero : i, n
inicio
   leer(n)
   i ← 2
   mientras i <= n hacer
      escribir(i)
      i ← i + 2
   fin_mientras
fin

//Versión utilizando estructura repetir
algoritmo NúmeroPares2
var
   entero : i, n
inicio
   leer(n)
   i ← 2
   repetir
      escribir(i)
      i ← i + 2
   hasta_que i > n
fin
```

**Puntuación: 1,5 puntos**

## Preguntas prácticas

Se desea obtener la tabla de puntuación en una liga de futbolín en la que compiten 10 equipos a una sola vuelta. Los resultados se almacenarán en una tabla de dos dimensiones, de forma que cada elemento i,j de la tabla guarda los tantos que ha marcado el equipo i al equipo j en el encuentro. Por ejemplo, un 8 en la posición 2,6 significa que en el encuentro entre el equipo 2 y el equipo 6, el equipo 2 ha metido 8 tantos al equipo 6.

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | | 3 | 6 | 2 | 5 | 6 | 4 | 2 | 0 | 2 |
| **2** | 3 | | 3 | 0 | 9 | 8 | 7 | 6 | 5 | 5 |
| **3** | 3 | 4 | | 7 | 8 | 9 | 6 | 5 | 4 | 3 |
| **4** | 5 | 6 | 2 | | 8 | 0 | 9 | 0 | 0 | 5 |
| **5** | 5 | 6 | 7 | 8 | | 3 | 5 | 4 | 3 | 2 |
| **6** | 0 | 2 | 3 | 3 | 4 | | 4 | 7 | 8 | 4 |
| **7** | 3 | 5 | 6 | 6 | 7 | 8 | | 3 | 8 | 9 |
| **8** | 3 | 5 | 7 | 1 | 2 | 0 | 4 | | 7 | 5 |
| **9** | 2 | 3 | 4 | 9 | 8 | 7 | 4 | 5 | | 1 |
| **10** | 3 | 4 | 5 | 6 | 5 | 6 | 7 | 4 | 4 | |

Los nombres de los equipos y la puntuación obtenida por cada uno de ellos se almacenarán en un array de registros de 10 posiciones (una por equipo) que almacenará el nombre del equipo y la puntuación.

Se pide:

- Declarar las estructuras de datos necesarias para realizar la aplicación.
- Diseñe un módulo que permita introducir por teclado el nombre de los equipos y almacenarlo en el array de registros.
- Diseñe un módulo que permita rellenar la tabla con los tantos obtenidos por los equipos.
- Diseñe un módulo que permita obtener la puntuación de cada equipo (3 puntos por partido ganado, 1 por partido empatado y 0 por partido perdido).

**Puntuación: 3,5 puntos**

### Declaraciones de las estructuras de datos

```
tipos
   registro : equipo
      cadena : nombre
      entero : puntos
   fin_registro
   array[1..10] de equipo: equipos
   array[1..10,1..10] de entero: tantos
```

### Introducir nombres de equipos

```
procedimiento IntroducirNombresEquipos(S equipos : e; E entero : n)
var
   entero : i
inicio
   desde i ← 1 hasta n hacer
      leer(e[i].nombre)
   fin_desde
fin_procedimiento
```

### Introducir tantos de los partidos

```
procedimiento IntroducirTantos(S tantos : t; E entero : n)
var
   entero : i,j
inicio
   desde i ← 1 hasta n - 1 hacer
      //Como el equipo i no juega con el equipo i, considero que la puntación es 0.
      //Esto facilitará el cálculo de la puntuación
      tantos[i,i] ← 0
      desde j ← i+1 hasta n hacer
         //Leer los tantos del equipo i contra el equipo j
         leer(t[i,j])
         //Leer los tantos del equipo j contra el equipo i
         leer(t[j,i])
      fin_desde
   fin_desde
fin_procedimiento
```

### Calcular la puntuación

```
procedimiento IntroducirTantos(E tantos : t; E/S equipos : e ; E entero : n)
var
   entero : i,j
inicio
   desde i ← 1 hasta n-1 hacer
      //Se inicializa a 0 los puntos del equipo i
      e[i].puntos ← 0
      desde j ← i+1 hasta n hacer
         si t[i,j] > t[j,i] entonces
            e[i].puntos ← e[i].puntos + 3
         si_no
            si t[i,j] = t[j,i] entonces
              e[i].puntos ← e[i].puntos + 1
              e[j].puntos ← e[j].puntos + 1
            si_no
              e[j].puntos ← e[j].puntos + 1
            fin_si
         fin_si
      fin_desde
   fin_desde
fin_procedimiento
```