



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRAL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

Soluciones propuestas

Teoría.

Subprogramas: tipos y diferencias entre ellos.

Apartados 5.2 y 5.3 del libro de texto y apuntes de clase

- Escribir los módulos necesarios para calcular el número de días entre dos fechas dadas (f_1 y f_2 ($f_1 \leq f_2$))

```
....
tipos
  registro
    a,m,d : entero
  fin_registro = fecha
....

lógico : función EsBisiesto(E entero : a)
inicio
  devolver((a mod 4 = 0) y ((a mod 100 <> 0) o (a mod 400 = 0)))
fin_función

//Cuenta los días transcurridos desde una fecha inicial a otra
//introducida como parámetro en forma de registro de tipo fecha
//Los valores del día, mes y año de la fecha inicial aparecen en forma
//de constantes dIni, mIni y aIni
entero : función ContarDias(E fecha : f)
const
  dIni = 1
  mIni = 1
  aIni = 1583 //Primer año del calendario gregoriano
var
  entero : conta,i
inicio
  //Contar los días de años completos
  desde i ← aIni hasta f.a - 1 hacer
    si EsBisiesto(i) entonces
      conta ← conta + 366
    si_no
      conta ← conta + 365
  fin_si
  fin_desde

//contar días de meses completos
desde i ← 1 hasta f.m -1 hacer
  según_sea i hacer
    4,6,9,11 : conta ← conta + 30
    2 : si EsBisiesto(f.a) entonces
      conta ← conta + 29
    si_no
      conta ← conta + 28
```



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRAL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

```

        fin_si
    si_no
        conta ← conta + 31
    fin_según
fin_desde

    devolver(conta + f.d)
fin_función

//Se supone que las fechas son correctas y que f2 es mayor que f1
entero : función DiferenciaDias(E fecha : f1,f2)
inicio
    devolver(ContarDias(f2)-ContarDias(f1))
fin_función
```

1,5 puntos

Teoría.

Indique los distintos métodos de ordenación por inserción que conozca.

Apartado 9.2.2. del libro de texto y apuntes de clase

Escribir el pseudocódigo del método Shell y para el array; (2,1,6,7,10,9,5) realizar una traza de mismo.

Apartado 9.2.4 del libro de texto

1,5 puntos

Problema

Por cada corredor de los 5000 que participan en un maratón (con 20 vueltas a un circuito) se almacenan: nombre, nacionalidad y tiempos en segundos. Defina las estructuras de datos y escriba los módulos necesarios para determinar:

Los 3 primeros corredores ganadores (oro, plata, bronce)

Las posiciones al final de la vuelta 10, de los corredores de una nacionalidad que se introduce por teclado, si esta tiene participantes en la maratón.

La nación ganadora, siendo esta la que tiene mejor media aritmética de los tiempos totales de los participantes de esa nacionalidad.

```

algoritmo Maratón
const
    NumPar = 5000
    NumVueltas = 20
tipos
    registro
        cadena : nom, pais
        array [1..NumVueltas] de real : t
    fin_registro = atleta
    array[0..NumPar] de atleta = carrera
    array[0..NumPar] de real = tiempos
    ...
    ...
```



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

```
//Obtener las medallas
procedimiento Medallas(E carrera : c; E entero : n)
var
    tiempos : tiemp
    entero : i
inicio
    //Obtener tiempos totales por atleta
    desde i ← 1 hasta n hacer
        tiempo[i] ← CalcularTiempo(c[i],NumVueltas)
    fin_desde
    Ordenar(c,tiemp,n)
    desde i ← 1 hasta 3 hacer
        escribir(c[i].nom)
    fin_desde
fin_procedimiento

//Calcula el tiempo de las n primera vueltas del atleta
real : función CalcularTiempo(E Atleta : a; E : entero : n)
var
    entero : conta, i
inicio
    conta ← 0
    desde i ← 1 hasta n hacer
        conta ← conta + a.t[i]
    fin_desde
    devolver(conta)
fin_función

//Obtener las posiciones de los paises en las tres primeras vueltas
procedimiento Posiciones(E carrera : c; E entero : n)
var
    tiempos : tiemp
    entero : i
    cadena : pais
inicio
    //Obtener tiempos totales por atleta
    desde i ← 1 hasta n hacer
        tiempo[i] ← CalcularTiempo(c[i],10)
    fin_desde
    Ordenar(c,tiemp,n)
    desde i ← 1 hasta n hacer
        si c[i].pais = pais entonces
            escribir(c[i].nom, i)
        fin_si
    fin_desde
fin_procedimiento

//Obtener el pais ganador
procedimiento PaisGanador(E carrera : c; E entero : n)
var
    tiempos : tiemp
```



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRAL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

```
entero : i
real : media,min
cadena : pais, paisAux
inicio
//Obtener tiempos totales por atleta
desde i ← 1 hasta n hacer
    tiemp[i] ← CalcularTiempo(c[i],NumVueltas)
fin_desde
OrdenarPorPais(c,tiemp,n)
min ← 0
i ← 1
mientras i < n hacer
    media ← 0
    j ← 0
    paisAux ← c[i].pais
    //Suma los tiempos del pais
    mientras paisAux = c[i].pais
        media ← media + tiemp[i]
        i ← i + 1
        j ← j + 1
    hasta_que c[i].pais <> paisAux o i = n
    si i = n entonces
        media ← media + tiemp[i]
        j ← j + 1
    fin_si
    media ← media / j
    si media < min entonces
        pais ← paisAux
        min ← media
    fin_si
fin_mientras
escribir('Pais ganador: ', pais)
fin_procedimiento
```

3,5 puntos

Problema

Dados dos numeros enteros introducidos por teclado realizar los siguientes procedimientos y/o funciones: Descomponer cada numero en factores primos. Almacenar cada factor, junto con su exponente en un nodo de una estructura simplemente enlazada. Cada lista contendrá la descomposición factorial de un número. Mostrar los factores primos de cada número en orden descendente de valor. Decir si hay más factores comunes que no comunes. Indicar el factor común con exponente más bajo, y el no común con exponente mas alto si existen.

```
tipos
registro
    entero : fact,exp
fin_registro = TipoElemento
```



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRAL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

```
puntero_a nodo = lista
registro
  TipoElemento : info
  lista : sig
fin_registro = nodo
```

```
procedimiento LInsertar(E/S lista : l; E TipoElemento:e)
var
```

```
  lista : aux
```

```
inicio
```

```
  reservar(aux)
  aux↑.info ← e
  aux↑.sig ← l
  l ← aux
```

```
fin_procedimiento
```

```
procedimiento descomponer(E entero : n; E/S lista : lista)
var
```

```
  entero : i, conta
  TipoElemento : e
  lista : aux
```

```
inicio
```

```
  //Inserta un nodo vacío y l apunta a él
  e.fact ← 0
  e.exp ← 0
  LInsertar(l,e)
  aux ← l
```

```
  i ← 2
```

```
  conta ← 0
```

```
  mientras I <= n hacer
```

```
    si n mod i = 0 entonces
```

```
      conta ← conta + 1
```

```
      n ← n div i
```

```
    si_no
```

```
      //Si se ha encontrado algún factor se inserta en la lista
```

```
      si conta > 0 entonces
```

```
        e.fact ← i
```

```
        e.exp ← conta
```

```
        //Se inserta después de aux para que la lista quede
```

```
        //ordenada de menor a mayor factor
```

```
        LInsertar(l,aux↑.sig)
```

```
        aux ← aux↑.sig
```

```
      fin_si
```

```
      conta ← 0
```

```
      i ← i + 1
```

```
    fin_si
```

```
  fin_mientras
```

```
  //Si hay algún factor más, se inserta
```



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRAL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

```
    si conta > 0 entonces
        e.fact ← i
        e.exp ← conta
        LInsertar(1, aux↑.sig)
    fin_si

    //Se borra el primer nodo vacío
    aux ← 1
    l ← l↑.sig
    liberar(aux)
fin_procedimiento

procedimiento MostrarDescendente(E lista : l)
inicio
    si l <> nulo entonces
        MostrarDescendente(l↑.sig)
        escribir(l↑.info.fact, l↑.info.exp)
    fin_si
fin_procedimiento

procedimiento NumComunesNoComunes(E lista : n1, n2)
var
    entero : Comunes, NoComunes
inicio
    Comunes ← 0
    NoComunes ← 0
    mientras (l1 <> nulo) y (l2 <> nulo) hacer
        si (l1↑.info.fact < l2↑.info.fact) entonces
            NoComunes ← NoComunes + 1
            l1 ← l1↑.sig
        si_no
            si (l1↑.info.fact > l2↑.info.fact) entonces
                NoComunes ← NoComunes + 1
                l2 ← l2↑.sig
            si_no
                Comunes ← Comunes + 1
                l1 ← l1↑.sig
                l2 ← l2↑.sig
        fin_si
    fin_si
fin_mientras
mientras (l1 <> nulo) hacer
    NoComunes ← NoComunes + 1
    l1 ← l1↑.sig
fin_mientras
mientras (l2 <> nulo) hacer
    NoComunes ← NoComunes + 1
    l2 ← l2↑.sig
```



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRAL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

```
    fin_mientras

    si Comunes > NoComunes entonces
        escribir('Más comunes que no comunes)
    si_no
        escribir('Más no comunes que comune)
    fin_si
fin_procedimiento

procedimiento MasAltoMasBajo(E lista : n1,n2)
var
    entero : Alto,Bajo
inicio
    Alto ← 11↑.info.exp
    Bajo ← 11↑.info.exp
    mientras (11 <> nulo) y (12 <> nulo) hacer
        si (11↑.info.fact < 12↑.info.fact) entonces
            si 11↑.info.exp > Alto entonces
                Alto ← 11↑.info.exp
            fin_si
            11 ← 11↑.sig
        si_no
            si (11↑.info.fact > 12↑.info.fact) entonces
                si 12↑.info.exp > Alto entonces
                    Alto ← 12↑.info.exp
                fin_si
                11 ← 11↑.sig
            si_no
                si 11↑.info.exp < Bajo entonces
                    Bajo ← 11↑.info.exp
                fin_si
                11 ← 11↑.sig
                12 ← 12↑.sig
            fin_si
        fin_si
    fin_si
fin_mientras

    mientras (11 <> nulo) hacer
        si 11↑.info.exp > Alto entonces
            Alto ← 11↑.info.exp
        fin_si
        11 ← 11↑.sig
    fin_mientras

    mientras (12 <> nulo) hacer
        si 12↑.info.exp > Alto entonces
            Alto ← 12↑.info.exp
```



Cuadernillo de examen

ASIGNATURA	FUNDAMENTOS DE PROGRAMACIÓN I	CÓDIGO	196
CONVOCATORIA	FEBRERO	PLAN DE ESTUDIOS	2001
ESPECIALIDAD		CURSO	1
TURNO	MAÑANA	CENTRO	ESCUELA
CARÁCTER	CUATRIMESTRL	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	3,0 HORAS		

```
    fin_si
    12 ← 12↑.sig
  fin_mientras

  si Comunes > NoComunes entonces
    escribir('Más comunes que no comunes)
  si_no
    escribir('Más no comunes que comune)
  fin_si
fin_procedimiento
```

3,5 puntos